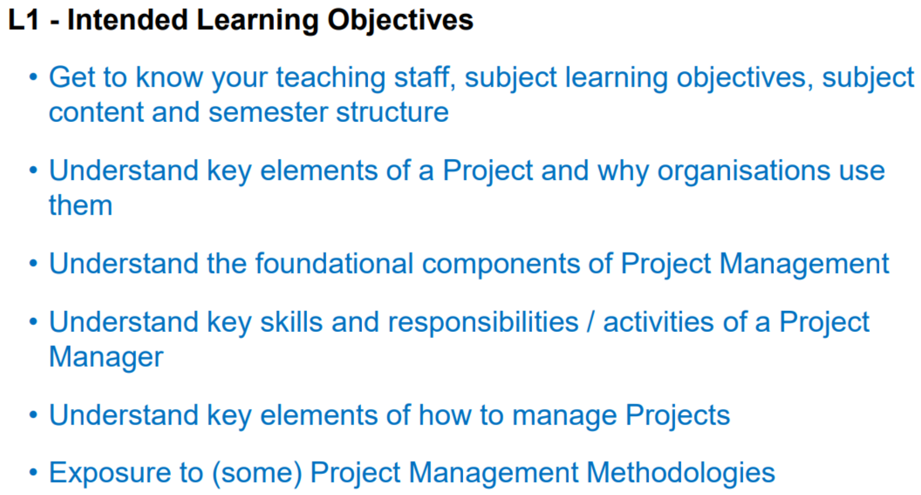
Software Processes & Project Management Notes

Lecture 1 (Introduction):



L1 - Intended Learning Objectives

- Get to know your teaching staff, subject learning objectives, subject content and semester structure
- Understand key elements of a Project and why organisations use them
- Understand the foundational components of Project Management
- Understand key skills and responsibilities / activities of a Project Manager
- Understand key elements of how to manage Projects
- Exposure to (some) Project Management Methodologies

Project: Temporary endeavor to create product/service or outcome. Introduced change to organization, has a start and end, varies in size.

Why projects: Gives structure, establishes ownership/accountability, provides strategic alignment and visibility. Efficient use of organization's limited resources.

Project management: Planning, delegating and monitoring all aspects of a project. Motivating those involved to achieve project objectives within expected time, cost, quality and benefit targets.

Skills/Attributes of PM:

- Organized
- Good at managing scarce resources and risk
- Take accountability, make project goals their own.
- Drive business results
- Use expertise and skill to inspire shared purpose of team

Key activities of a PM:

Planning: Define and clarify project scope, make a PM plan, create procedures

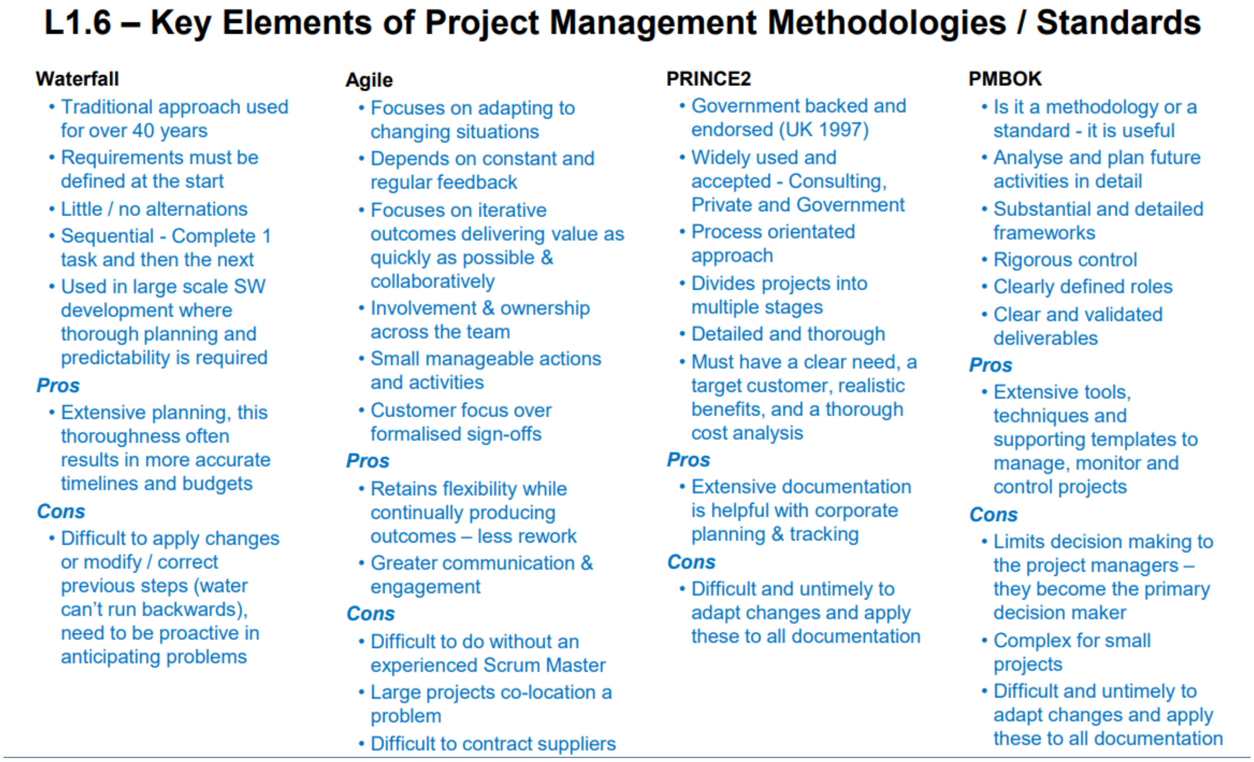
Organizing: Determine team structure, identify roles, staff all positions also identifying services needed by external companies.

Leading: Setting team direction, owning and coordinating activities, motivating, assigning work.

Controlling: Defining goals, tracking and reporting progress/status.

A change is occurring in PM:

- Agile changing the role of a PM – No define PM role.
- Key activities spread/shared across team
- Some alignment to activities of scrum master
- Motivate team, emphasis objectives, invest in performance, question, assist.



L1.6 – Key Elements of Project Management Methodologies / Standards

Waterfall

- Traditional approach used for over 40 years
- Requirements must be defined at the start
- Little / no alternations
- Sequential - Complete 1 task and then the next
- Used in large scale SW development where thorough planning and predictability is required

Pros

- Extensive planning, this thoroughness often results in more accurate timelines and budgets

Cons

- Difficult to apply changes or modify / correct previous steps (water can't run backwards), need to be proactive in anticipating problems

Agile

- Focuses on adapting to changing situations
- Depends on constant and regular feedback
- Focuses on iterative outcomes delivering value as quickly as possible & collaboratively
- Involvement & ownership across the team
- Small manageable actions and activities
- Customer focus over formalised sign-offs

Pros

- Retains flexibility while continually producing outcomes – less rework
- Greater communication & engagement

Cons

- Difficult to do without an experienced Scrum Master
- Large projects co-location a problem
- Difficult to contract suppliers

PRINCE2

- Government backed and endorsed (UK 1997)
- Widely used and accepted - Consulting, Private and Government
- Process orientated approach
- Divides projects into multiple stages
- Detailed and thorough
- Must have a clear need, a target customer, realistic benefits, and a thorough cost analysis

Pros

- Extensive documentation is helpful with corporate planning & tracking

Cons

- Difficult and untimely to adapt changes and apply these to all documentation

PMBOK

- Is it a methodology or a standard - it is useful
- Analyse and plan future activities in detail
- Substantial and detailed frameworks
- Rigorous control
- Clearly defined roles
- Clear and validated deliverables

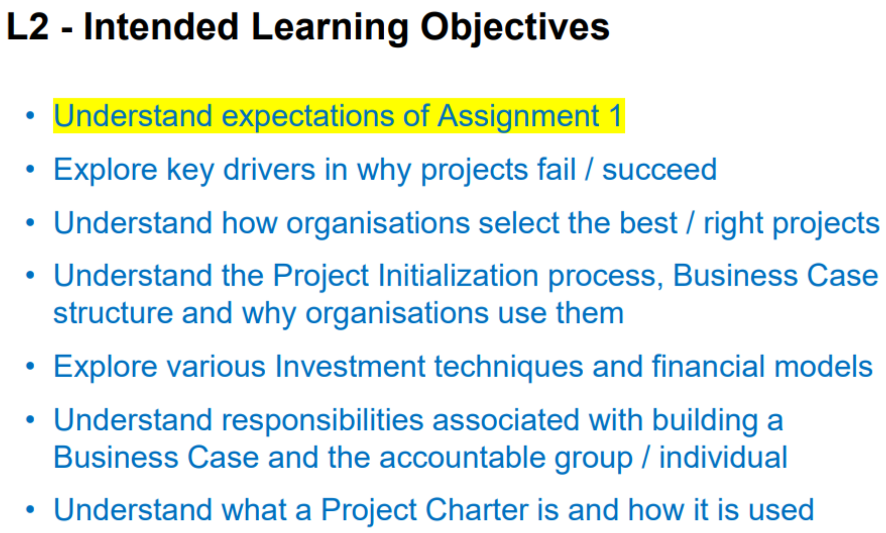
Pros

- Extensive tools, techniques and supporting templates to manage, monitor and control projects

Cons

- Limits decision making to the project managers – they become the primary decision maker
- Complex for small projects
- Difficult and untimely to adapt changes and apply these to all documentation

Lecture 2 (Project Initialization and Business Drivers):



L2 - Intended Learning Objectives

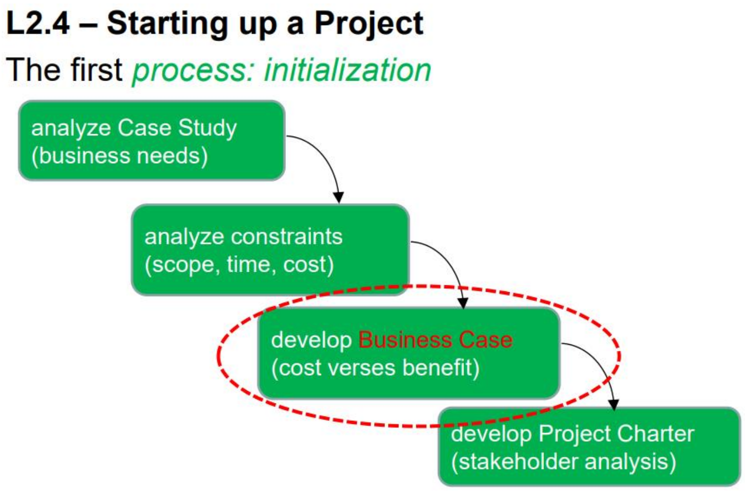
- Understand expectations of Assignment 1
- Explore key drivers in why projects fail / succeed
- Understand how organisations select the best / right projects
- Understand the Project Initialization process, Business Case structure and why organisations use them
- Explore various Investment techniques and financial models
- Understand responsibilities associated with building a Business Case and the accountable group / individual
- Understand what a Project Charter is and how it is used

Key Drivers that determine Success/Failure of Project:

- Executive sponsorship
- Emotional maturity
- User involvement
- Optimisation – Statement of Requirements

Project Initialization:

- PRINCE2
- PMBOK



L2.4 – Starting up a Project

The first *process: initialization*

analyze Case Study (business needs)

analyze constraints (scope, time, cost)

develop Business Case (cost verses benefit)

develop Project Charter (stakeholder analysis)

Business case: Purpose of business case is to establish mechanisms to judge whether the project is desirable, viable and achievable as a means to support decision making in initial and continued investment.

Business case is key and contains:

- Executive summary
- Purpose
- Business options
- Benefits
- Time
- Cost
- Risk

Determines if project should be undertaken and how it adds value.

Investment Aspects too look in Project:

- Return on interest = (benefits – costs) / costs
- Net Present value: expected monetary gain or loss from investment by discounting all future costs and benefits to the present time.
- Payback period: Amount of time it takes a project before the accrued benefits surpass accrued costs. Time to recover initial cost.

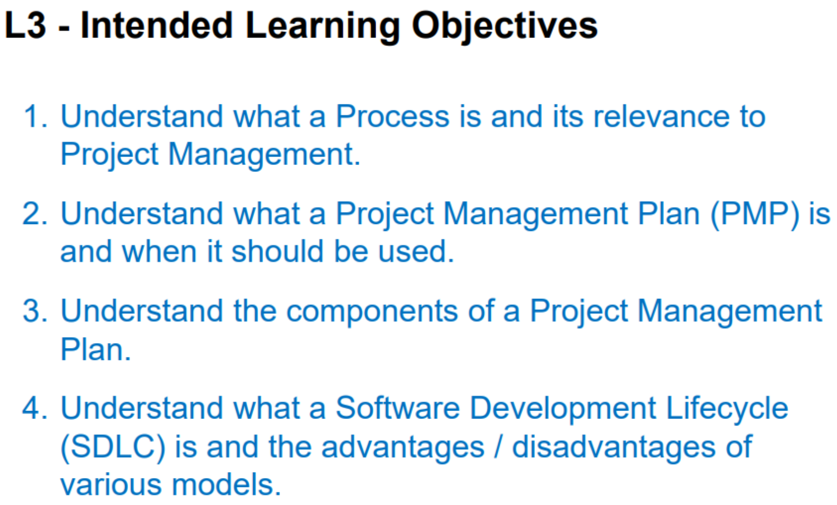
Responsibility of a PM:

1. Prepares business case
2. Conducts risk assessment and impact analysis
3. Assess and updates the business case at each defined stage

What is a Project Charter?

- Project description
- Costs of different resources based on quantity and rate
- Gains in project based on quantity and rate
- Project team
- Milestones.

Lecture 3 (PMP and SDLC):



L3 - Intended Learning Objectives

1. Understand what a Process is and its relevance to Project Management.
2. Understand what a Project Management Plan (PMP) is and when it should be used.
3. Understand the components of a Project Management Plan.
4. Understand what a Software Development Lifecycle (SDLC) is and the advantages / disadvantages of various models.

What is a Process: A series of progressive and interdependent steps by which an end is attainted.

What does a process have to do with PM and software engineering:

- PM is a process as it defines a series of tasks (Planning, controlling) to deliver a specific/agreed outcome.
- System Development Lifecycle (SDLC) describes process for planning, creating, testing and deploying an info system.

Project Management Plan:

- Formal approved document that defines how the project is executed, monitored and controlled. May be summary or detailed.
- Owned, controlled and populated by PM.
- A good PMP provides required level of detail across key project components and is the one source of truth for all parties involved in project.

Difference between project charter and PMP:

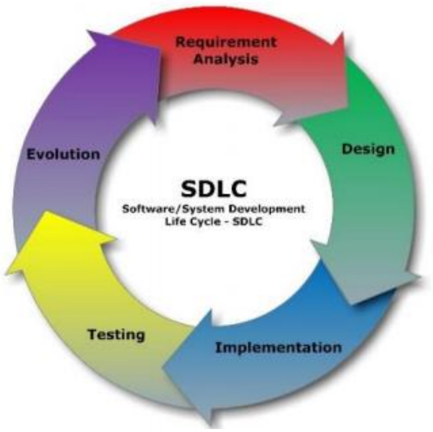
- A Project charter is a summary project proposal to secure approval for the project goals and terms. (useful for business case)
- PMP is approved document showing how to achieve the approved project goals/benefits and provides details on how to execute and manage project.

Things in PMP:

- Executive summary
- Scope
- Milestones
- Stakeholders
- Business values
- Budget + cost estimation
- Constraints
- Roles and responsibilities
- Schedule
- Risk management
- Quality assurance

SDLC: Systems Development Life Cycle

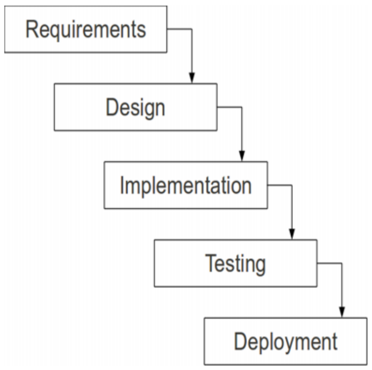
- Process for planning, creating testing and deploying and info system.
- Key activities displayed in diagram.
- Formal Processes:
    - Waterfall
    - Incremental
    - V-Model
- Agile Processes:
    - Scrum
    - Kanban



Waterfall: Requirements first driven approach

Advantages:

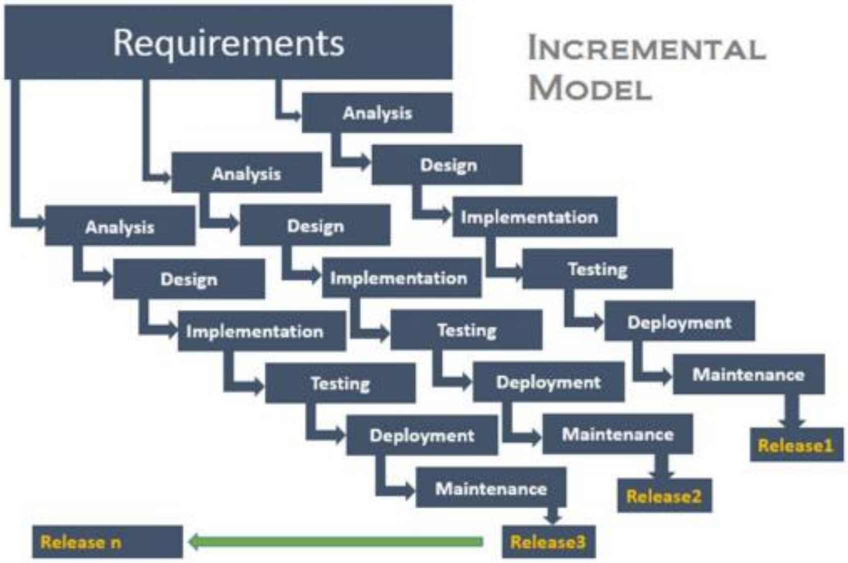
- Works well where requirements well understood and are stable
- Simple and easy to understand/use
- Easy manage due to rigidity of model
- Phases complete one at a time
- Documentation available at end of each phase



Disadvantages:

- Difficult to accommodate change after the process is underway.
- One phase must be completed before moving onto the next.
- Unclear requirements lead to confusion.
- Clients approval is in the final stage.
- Difficult to integrate risk management due to uncertainty.

Incremental: The whole requirements is divided into various releases. Multiple cycles take placing making the life cycle a multi-waterfall cycle. Cycles are divided into smaller more easily managed modules. Still getting requirements upfront.



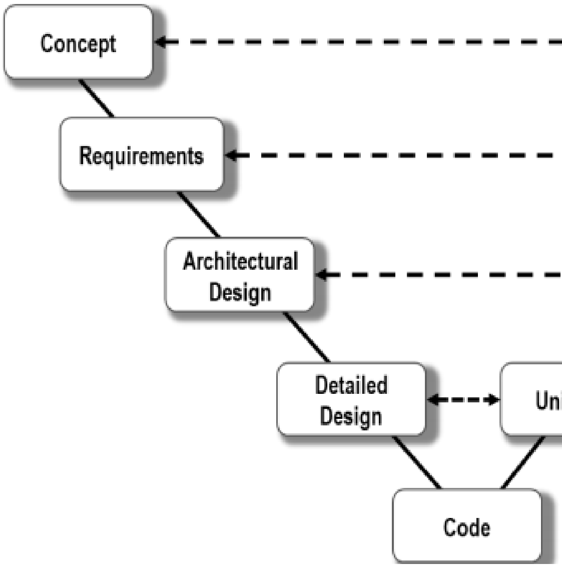
Advantages:

- Each release delivers an operational product.
- Less costly to change the scope/requirements
- Customers can respond to each build
- Initial product deliver is faster
- Customers get important functionality early
- Easier to test and debug during smaller iterations

Disadvantages:

- More resources and management attention required
- Defining/partitioning increments may be difficult/unclear
- Each phase of iteration is rigid with no overlaps
- Problems may occur at the time of final integration

V Model: Extension of the waterfall model and is based on the association of a testing phase for each corresponding development stage. Every single phase in the development cycle has a directly associated testing phase. The next phase starts only after the completion of the previous phase.



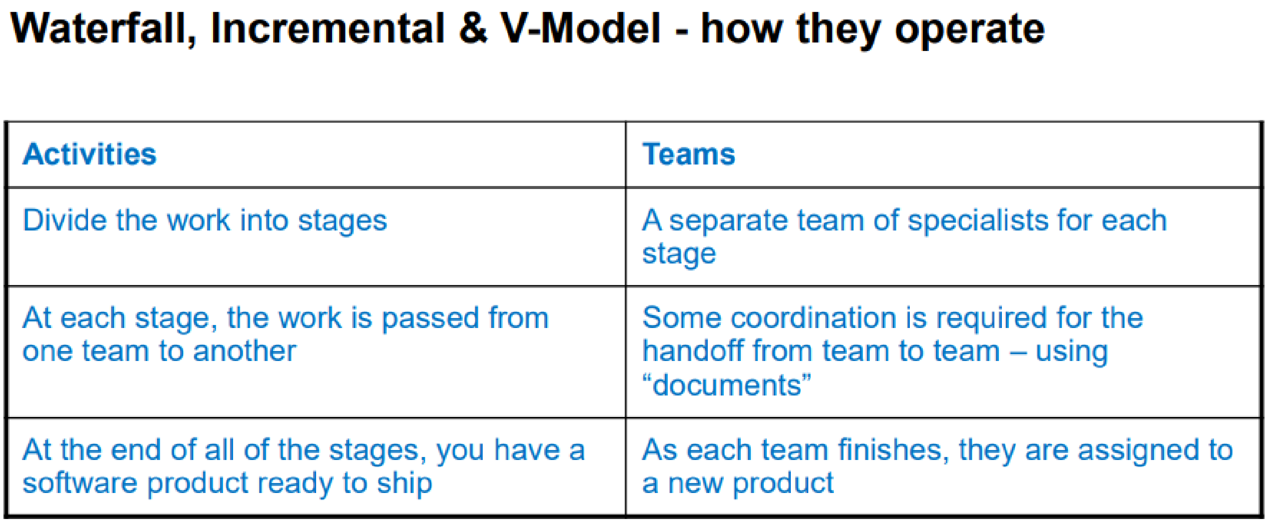
Advantages:

- Simple and easy to understand/use
- Each phase has specific deliverables and well-defined objectives/goals
- High chance of success over waterfall model due to the development of test plans early on during the life cycle.
- Works well for small projects when requirements easily understood.

Disadvantages:

- Very rigid process like the waterfall model
- Little flexibility and adjusting scope is difficult and expensive
- Software developed during implementation phase, no early prototypes
- No clear path for problems found during testing
- Changes in later stages cause test documentation across all stages to be changed.

Scope: requirements specification or product backlog.



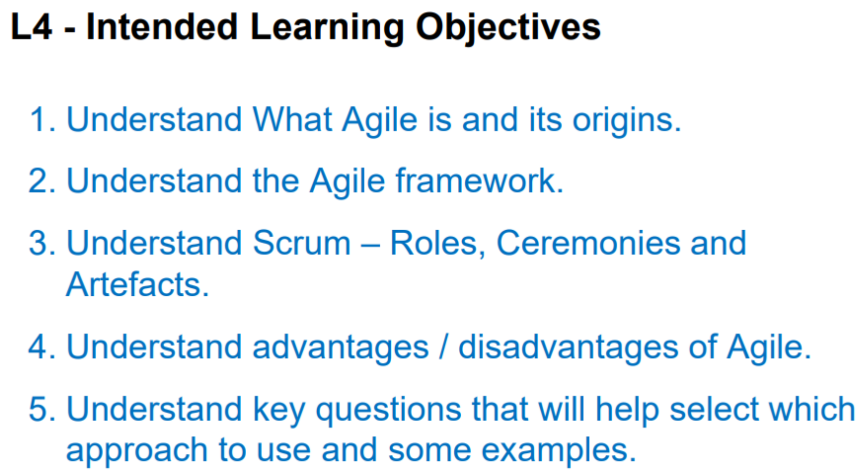
Waterfall, Incremental & V-Model - how they operate

Activities	Teams
Divide the work into stages	A separate team of specialists for each stage
At each stage, the work is passed from one team to another	Some coordination is required for the handoff from team to team – using “documents”
At the end of all of the stages, you have a software product ready to ship	As each team finishes, they are assigned to a new product

When to use these formal models?

- When customer has clear view of what they want.
- Project requires little or no change to requirements.
- Software requirements are clearly defined and documented.
- Tech and tools are well-known. Large scale apps and systems developments.

Lecture 4 and 5 (SDLC and Agile Scrum):



L4 - Intended Learning Objectives

1. Understand What Agile is and its origins.
2. Understand the Agile framework.
3. Understand Scrum – Roles, Ceremonies and Artefacts.
4. Understand advantages / disadvantages of Agile.
5. Understand key questions that will help select which approach to use and some examples.

Why is Agile attractive:

- Customer needs and demands exponentially increasing. Products need to be delivered
- Low tech cost, ease of use, increased competition
- Quality must be maintained

What is Agile:

- Set of methodologies based on iterative development where requirements and solutions evolve through collaboration between self-organizing cross-functional groups.
- Disciplined process that encourages frequent inspection and adaptation.
- Leadership philosophy that encourages teamwork, self-organization and accountability.
- Aligns development with customer needs and company goals.

Agile Manifesto:

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan

Agile Key principles:

- Satisfy customer through early and continuous deliver of software.
- Welcome changing requirements, even in late development.
- Deliver working software frequently. Delivery measures success.
- Business people + developers must work daily
- Sustainable development.
- Continuous attention to detail, enhances agility and design.
- Team reflects how to become better at regular intervals.

Agile Frameworks:

Kanban:

Work items visualized to provide participants a view of progress and process from start to finish via a Kanban board. To do, doing and done.

Scrum:

- Agile process that allows us to focus on deliver highest business values in shortest time.
- Allows us to rapidly and repeatedly inspect actual working software.
- The business sets the priorities. Team self-organize to determine the best way to deliver the highest priority features.
- Every two to four weeks, you can see real working software and decide to release it as is or continue to enhance it for another sprint.

Scrum Framework:

Roles:

- Product owner: Defines features of product. Makes product profitable and prioritizes features. Accepts or rejects work results.
- Scrum master: Represents management to the project. Enacts scrum values. Removes road blocks. Maintain functionality. Shield team from external.
- Team: typically, 5-9 people. Programmers, testers, user experience, BA.

Ceremonies:

Sprint Planning:

- Defines how to achieves print goal (design). Consider high level design.
- Create sprint backlog (user stories) from product backlog
- Estimate sprint backlog in team velocity and Story Points

Daily Stand-up:

- Daily, 15 minutes and standing up
- What I did yesterday, What I will do today, what is blocking me
- Avoid necessary meetings and get team together

Sprint Reviews – showcase:

- Team presents what it accomplished. Whole team participates.
- Demo of new features – informal.
- Chance for feedback, making team aware of what other worked on.

Sprint Retrospective:

- 30 minutes, whole team participates.
- What to start doing, stop doing and continue doing
- Don't after every sprint

Artifacts:

Product backlog:

- The requirements
- Lost of desired work on project
- Each item should have value to the users or customers of product
- Reprioritizes at start of each sprint.
- User stories: "As a site visitor, I can read current news on the home page"

Sprint backlog:

- Scum team decomposes to low level user stories during spring plan.
- User stories estimated in story points.
- Turn this story into a set of tasks with hours + estimates.

Burndown charts:

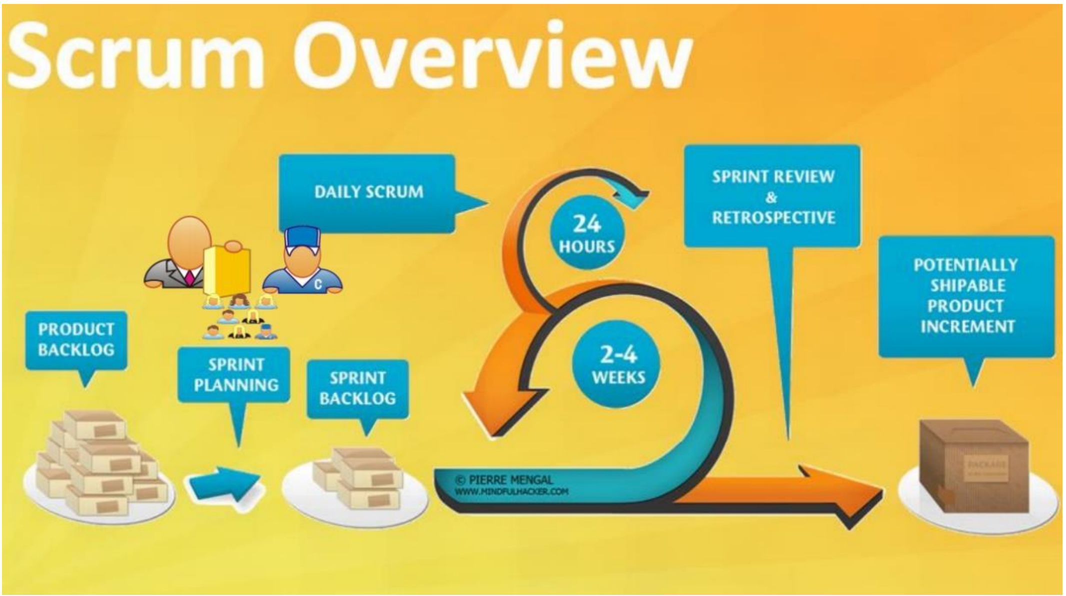
- Graphical representation of work left to do vs time.
- Outstanding work is over on the vertical axis, time horizontal.
- Used to predicted when all work will be done.

Agile advantages:

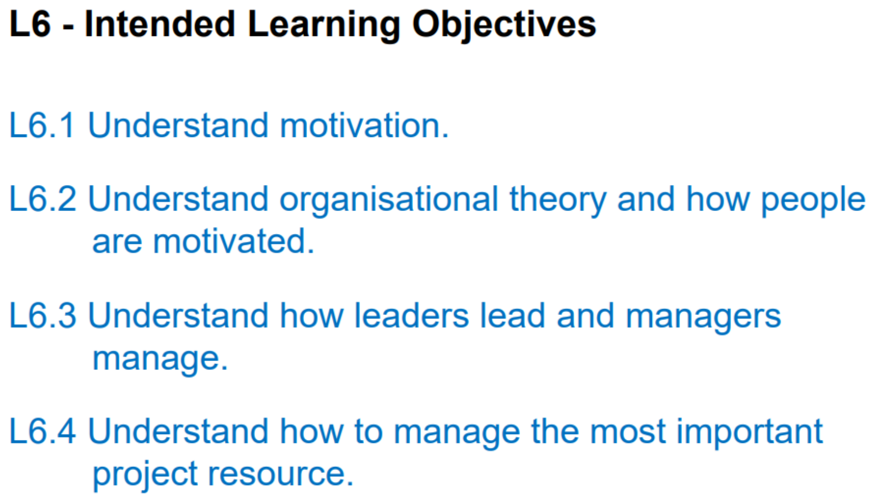
- Customer satisfaction by rapid continuous delivery of working software.
- People and interactions are emphasized rather than process and tools.
- Continues attention to technical excellence, good design + quality.
- Regular adaptation to changing circumstances.

Agile Disadvantages:

- Difficult to assess the effort required at beginning.
- Can be very demanding on user's times.
- Requires experienced resources.
-


Scrum Overview
DAILY SCRUM
24 HOURS
SPRINT REVIEW & RETROSPECTIVE
POTENTIALLY SHIPABLE PRODUCT INCREMENT
PRODUCT BACKLOG
SPRINT PLANNING
SPRINT BACKLOG
2-4 WEEKS
© PIERRE MENGAL WWW.MINDFULHACKER.COM

Lecture 6 (Individuals and Motivation):



L6 - Intended Learning Objectives

L6.1 Understand motivation.

L6.2 Understand organisational theory and how people are motivated.

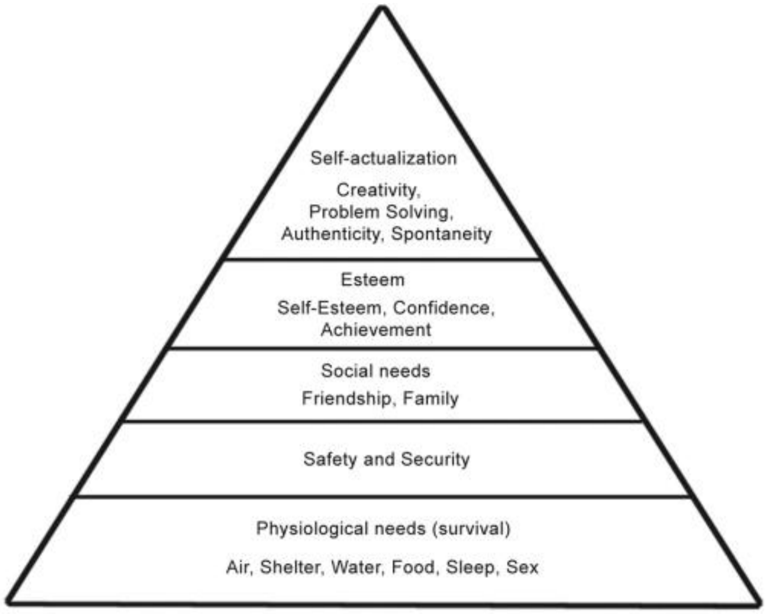
L6.3 Understand how leaders lead and managers manage.

L6.4 Understand how to manage the most important project resource.

Organizational Theory: Organizations are groups of individuals, structured and managed to meet a need, or to pursue collective goals.

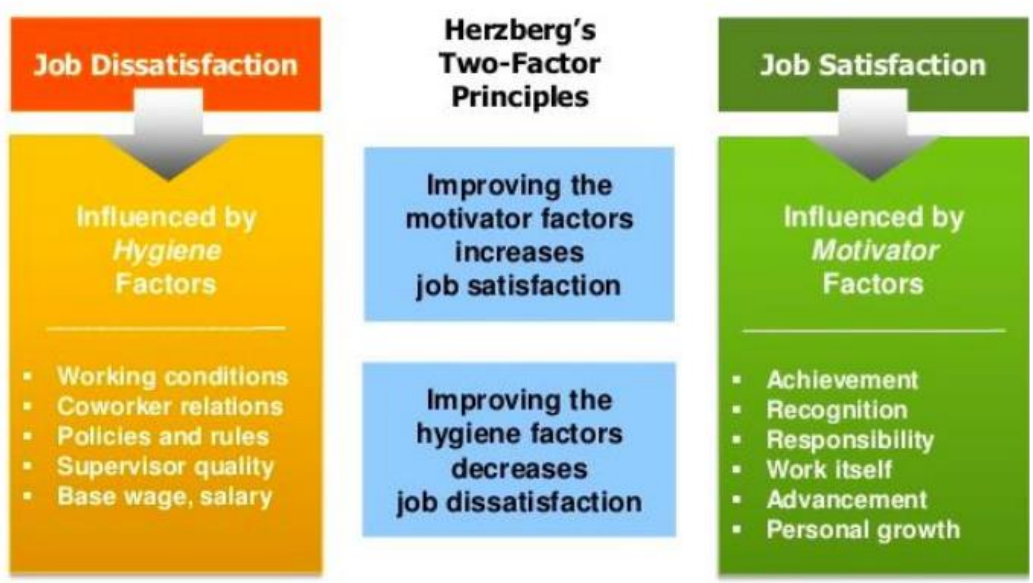
Maslow Hierarchy of Needs:

Describes how humans are motivated to achieve certain needs. Lower needs need to be met before higher needs.


Self-actualization
Creativity, Problem Solving, Authenticity, Spontaneity
Esteem
Self-Esteem, Confidence, Achievement
Social needs
Friendship, Family
Safety and Security
Physiological needs (survival)
Air, Shelter, Water, Food, Sleep, Sex

Hertzberg Two Factor Theory:

Remedying the causes of dissatisfaction will not create satisfaction. Adding job satisfaction will eliminate job dissatisfaction.



Herzberg's Two-Factor Principles

Job Dissatisfaction

Influenced by *Hygiene* Factors

- Working conditions
- Coworker relations
- Policies and rules
- Supervisor quality
- Base wage, salary

Improving the motivator factors increases job satisfaction

Improving the hygiene factors decreases job dissatisfaction

Job Satisfaction

Influenced by *Motivator* Factors

- Achievement
- Recognition
- Responsibility
- Work itself
- Advancement
- Personal growth

How leaders lead and manage:

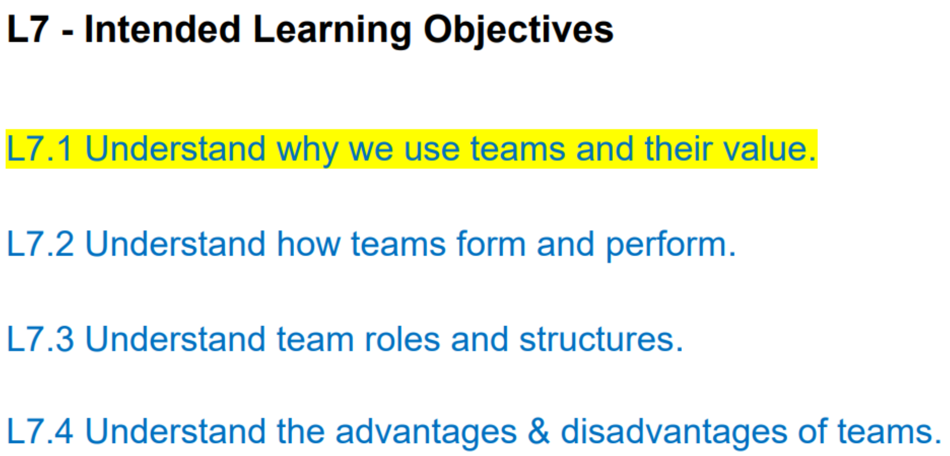
Power:

Expert power, reward power, coercive power (punishment).

Influence:

Authority, Money, Expertise, Friendship.

Lecture 7 (People and Teams):



L7 - Intended Learning Objectives

L7.1 Understand why we use teams and their value.

L7.2 Understand how teams form and perform.

L7.3 Understand team roles and structures.

L7.4 Understand the advantages & disadvantages of teams.

Team: A Team is two or more individuals consciously working together to achieve a common objective.

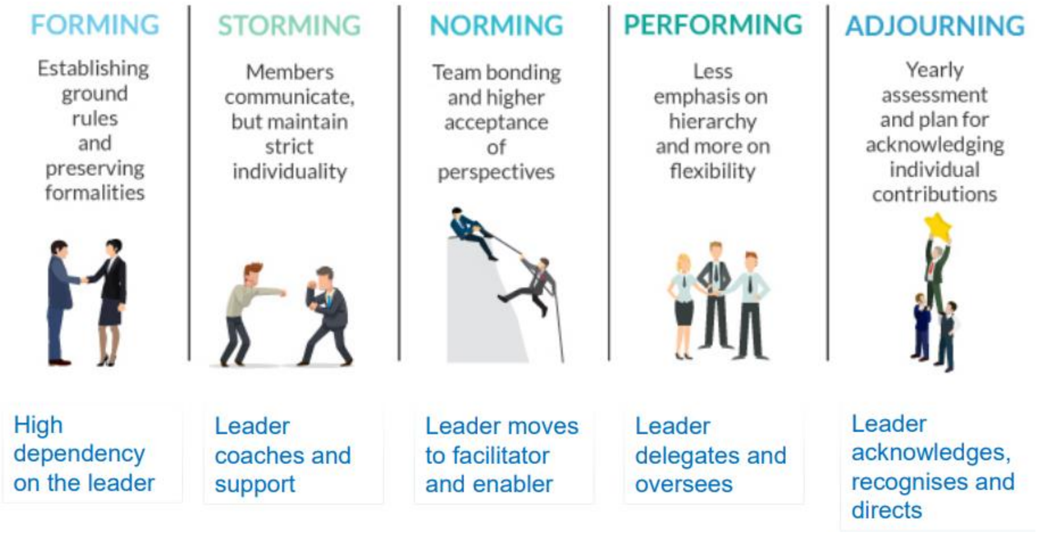
Benefits of Teams:

- Few individuals possess all the knowledge, skills and abilities to do all tasks.
- Complementary skills improve efficiency.
- Benefits to organization of team members.
- Shared accountability.
- More innovation.

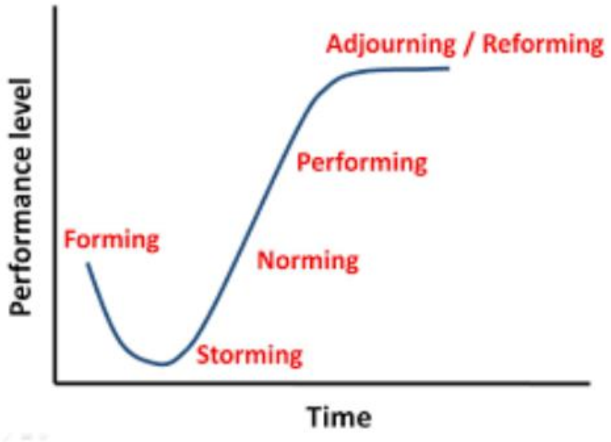
Sign that Team is Effective:

- Clear communication
- Consensus
- Commitment to project outcomes and each other
- Regular team meetings are effective and inclusive
- Clear roles/responsibilities
- Support for each other

Tuckman’s Team Development Model:



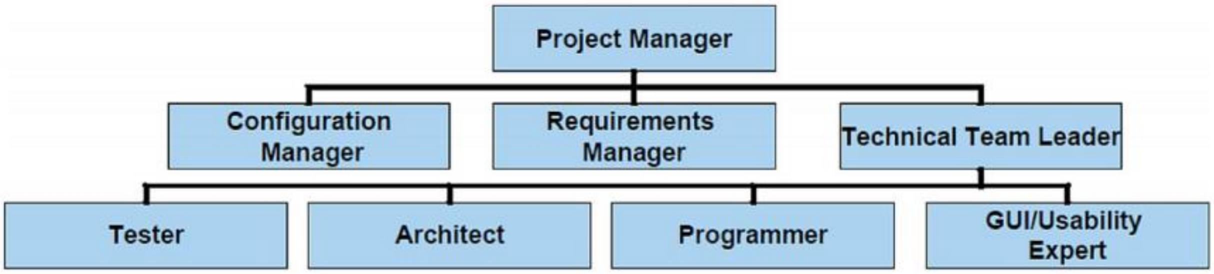
FORMING	STORMING	NORMING	PERFORMING	ADJOURNING
Establishing ground rules and preserving formalities	Members communicate, but maintain strict individuality	Team bonding and higher acceptance of perspectives	Less emphasis on hierarchy and more on flexibility	Yearly assessment and plan for acknowledging individual contributions
High dependency on the leader	Leader coaches and support	Leader moves to facilitator and enabler	Leader delegates and oversees	Leader acknowledges, recognises and directs



Team Roles and Structures:

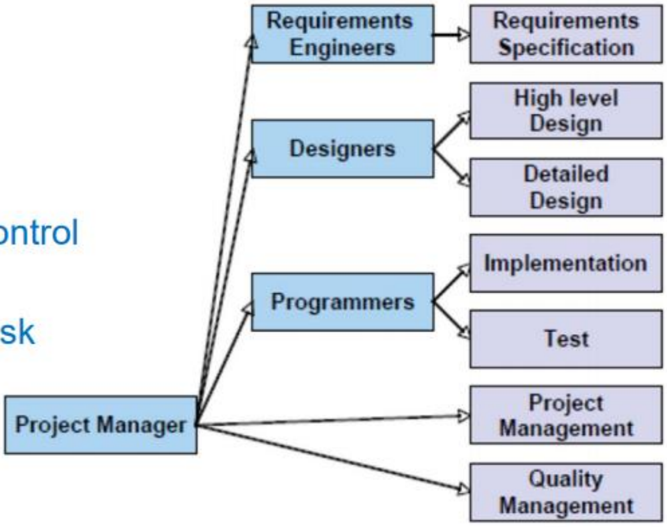
Controlled Centralised

- Leader coordinates tasks and directs work
- Communication and Control are vertical
- Sub-teams with leaders to direct and guide sub-groups



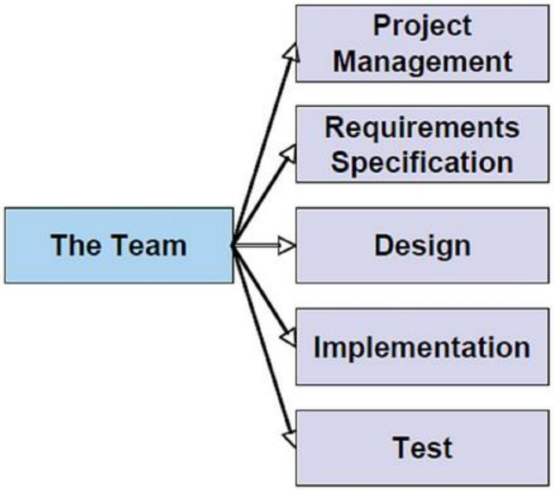
Controlled Decentralised

- Leader coordinates tasks
- Sub-teams with leaders
- Communication horizontal Control is vertical
- Problem solving is a team task



Democratic Decentralised

- No permanent leader
- A coordinator is appointed for short durations
- Communication and Control are horizontal

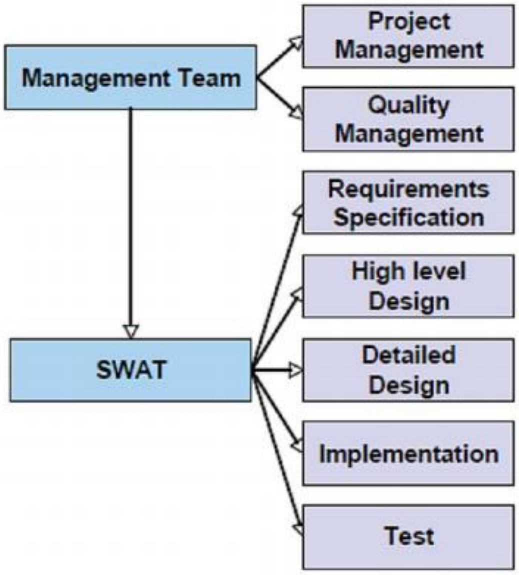


SWAT Team

- No permanent leader
- A highly specialised team
- Put together for a particular tasks

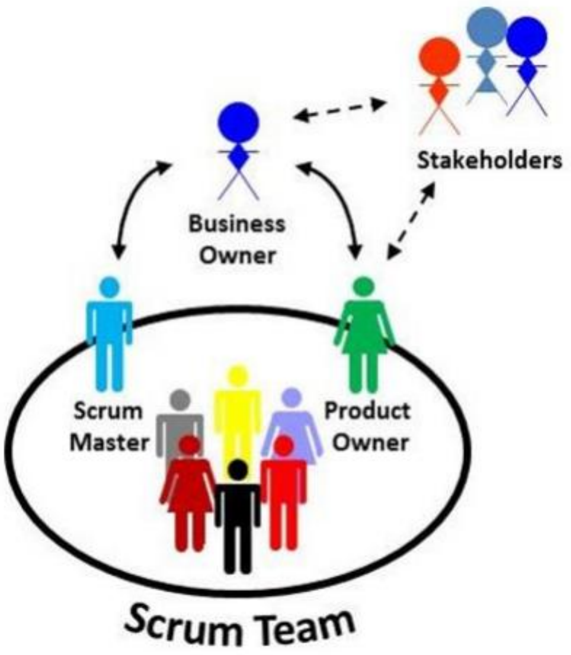
  e.g. - Build a prototype,

  - Production problem



Scrum Team

- Used in Agile



Team Advantages:

- Pool of ideas /creativity
- Interaction enhances knowledge of whole team
- Motivation
- More ownership

Team Disadvantages:

- Time, effort and skill to manage
- Unequal involvement
- Too many ideas/conflicts

Lecture 8 (Communication Management):

L8.1 Understand the communication challenge

L8.2 Understand the importance of listening

L8.3 Understand the importance of communicating effectively & some key skills

L8.4 Understand a communications plan and how it is used

Communication Challenges: Culture difference, Anxiety, status, location remote?

Importance of Listening:

- Promotes problem solving
- Demonstrates acceptance
- Shows respect
- Helps you understand and retain and help

Types of Listening:

Passive: Lectures – take in information.

Active/Empathetic: Tutorials: show interest, ask questions, eye contact

Critical: Challenge and question ideas, distinguish fact from opinion

Challenges: Distracted, no background for information, selective memory.

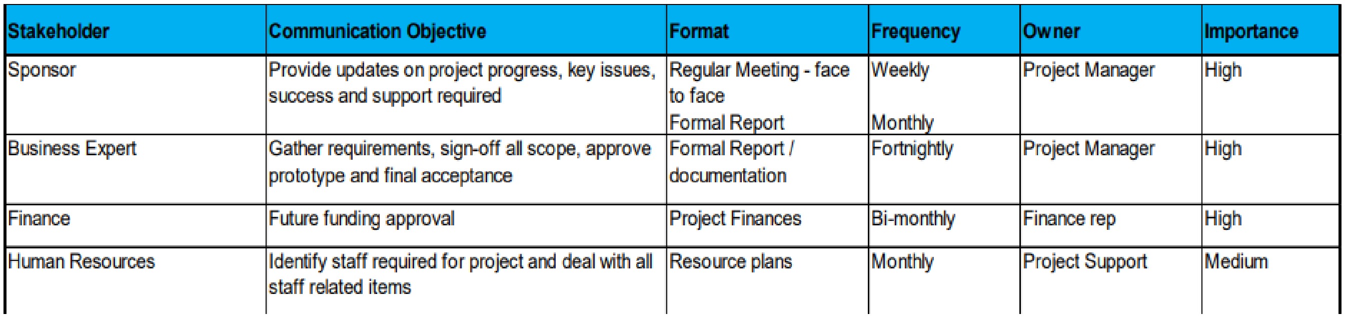
Communication Important for PM:

Motivating/influencing others, managing conflict, networking, delegating task, helping solve problems.

Communication Plan:

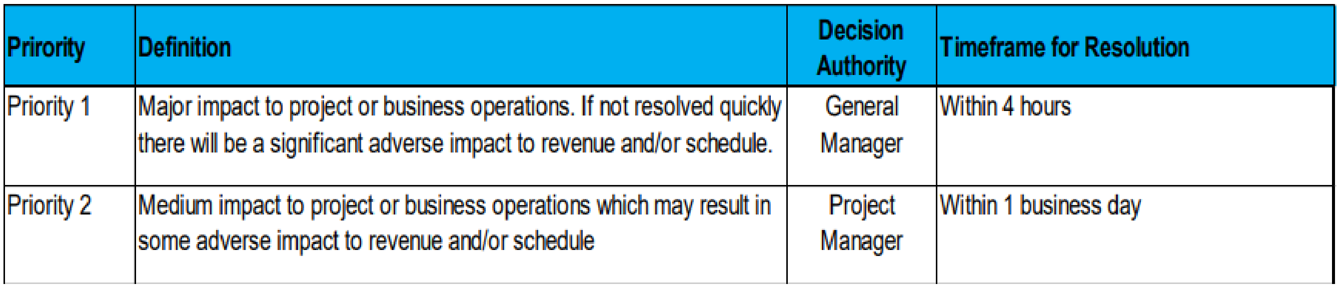
- Defines what info will be communicated and when
- Who is responsible
- Importance of communication

Communications Matrix:



Stakeholder	Communication Objective	Format	Frequency	Owner	Importance
Sponsor	Provide updates on project progress, key issues, success and support required	Regular Meeting - face to face Formal Report	Weekly Monthly	Project Manager	High
Business Expert	Gather requirements, sign-off all scope, approve prototype and final acceptance	Formal Report / documentation	Fortnightly	Project Manager	High
Finance	Future funding approval	Project Finances	Bi-monthly	Finance rep	High
Human Resources	Identify staff required for project and deal with all staff related items	Resource plans	Monthly	Project Support	Medium

Communication Escalation Plan:



Prirority	Definition	Decision Authority	Timeframe for Resolution
Priority 1	Major impact to project or business operations. If not resolved quickly there will be a significant adverse impact to revenue and/or schedule.	General Manager	Within 4 hours
Priority 2	Medium impact to project or business operations which may result in some adverse impact to revenue and/or schedule	Project Manager	Within 1 business day

Lecture 9 (Stakeholder Management):

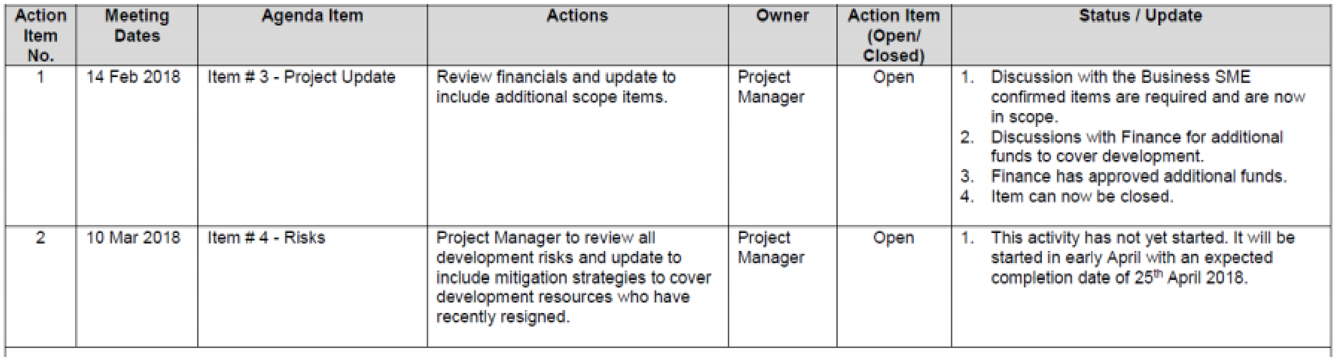
L8.5 Understand key communication considerations

L9.1 Understand the Stakeholder Management Process

L9.2 Identifying Stakeholders & the Stakeholder Register

L9.3 Understand Stakeholder Engagement and Planning

Action Register:



Action Item No.	Meeting Dates	Agenda Item	Actions	Owner	Action Item (Open/ Closed)	Status / Update
1	14 Feb 2018	Item # 3 - Project Update	Review financials and update to include additional scope items.	Project Manager	Open	1. Discussion with the Business SME confirmed items are required and are now in scope. 2. Discussions with Finance for additional funds to cover development. 3. Finance has approved additional funds. 4. Item can now be closed.
2	10 Mar 2018	Item # 4 - Risks	Project Manager to review all development risks and update to include mitigation strategies to cover development resources who have recently resigned.	Project Manager	Open	1. This activity has not yet started. It will be started in early April with an expected completion date of 25th April 2018.

Conflict can Occur across project life cycle:

1. **Definition/Mobilisation phase:** Which SDLC to use, importance of project
2. **Planning Phase:** Decision rights, technical solutions/stack, availability
3. **Execution Phase:** Missed milestones, 3rd party supplier issues
4. **Delivery Phase:** Scope changes, missed milestone, burnout

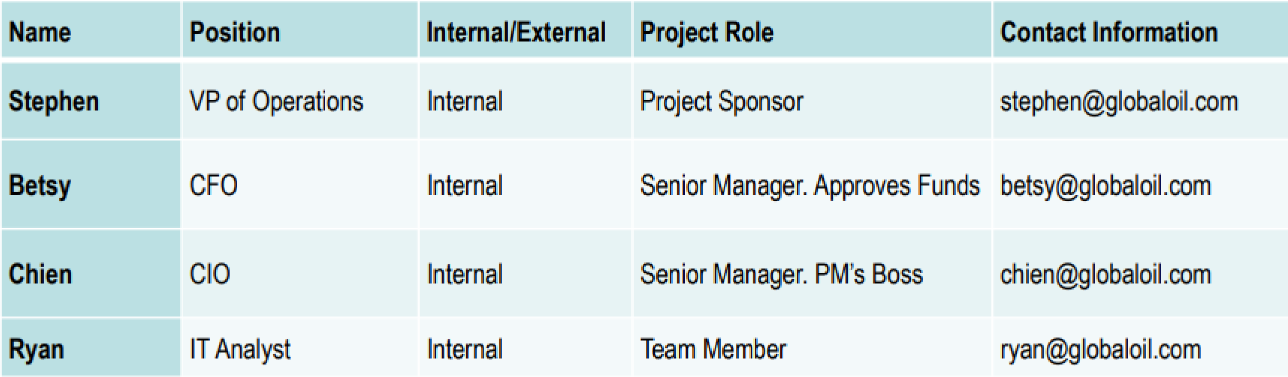
Identifying Stakeholders:

-Identify stakeholder's names, positions, locations, roles

-Classify stakeholder: internal/external/supporter/resistor.

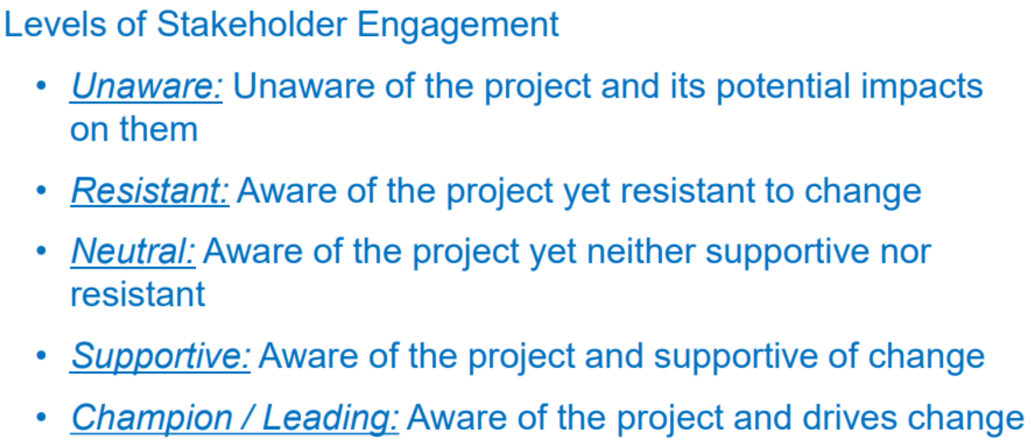
-The stakeholder's major requirements and expectations. Potential Influences.

Stakeholder Register:



Name	Position	Internal/External	Project Role	Contact Information
Stephen	VP of Operations	Internal	Project Sponsor	stephen@globaloil.com
Betsy	CFO	Internal	Senior Manager. Approves Funds	betsy@globaloil.com
Chien	CIO	Internal	Senior Manager. PM's Boss	chien@globaloil.com
Ryan	IT Analyst	Internal	Team Member	ryan@globaloil.com

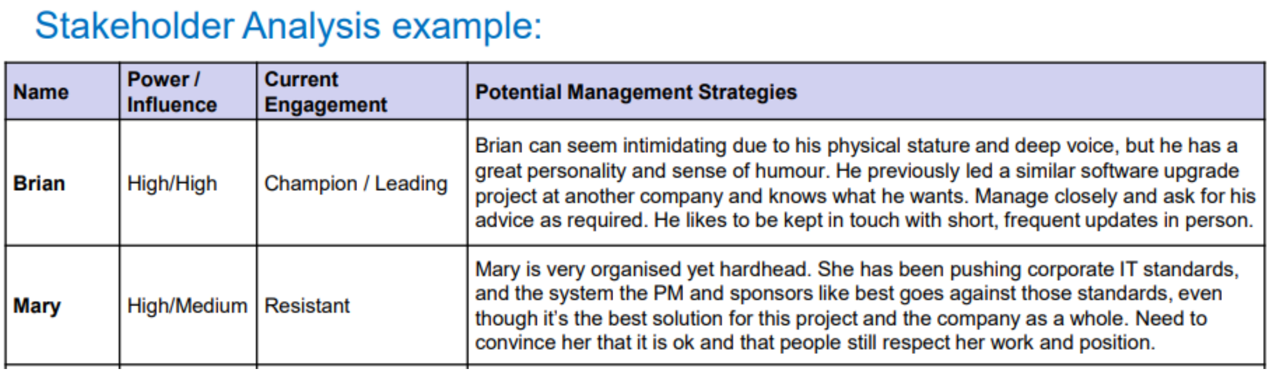
Understanding Stakeholder:



Levels of Stakeholder Engagement

- *Unaware:* Unaware of the project and its potential impacts on them
- *Resistant:* Aware of the project yet resistant to change
- *Neutral:* Aware of the project yet neither supportive nor resistant
- *Supportive:* Aware of the project and supportive of change
- *Champion / Leading:* Aware of the project and drives change

Strategies for managing/role on project/influence on project/level of interest



Stakeholder Analysis example:

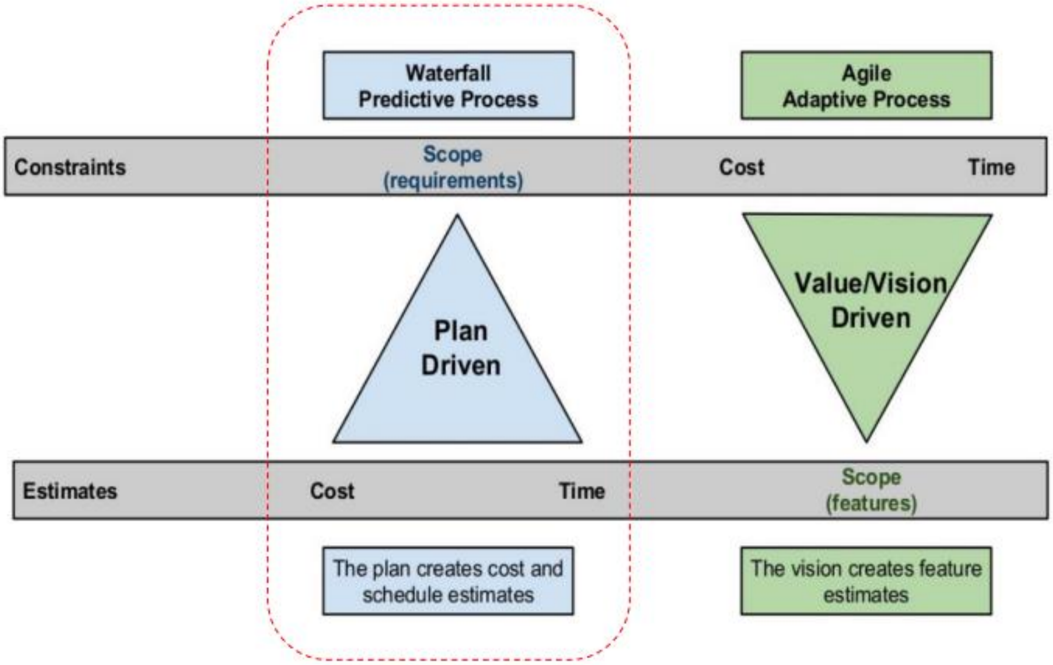
Name	Power / Influence	Current Engagement	Potential Management Strategies
Brian	High/High	Champion / Leading	Brian can seem intimidating due to his physical stature and deep voice, but he has a great personality and sense of humour. He previously led a similar software upgrade project at another company and knows what he wants. Manage closely and ask for his advice as required. He likes to be kept in touch with short, frequent updates in person.
Mary	High/Medium	Resistant	Mary is very organised yet hardhead. She has been pushing corporate IT standards, and the system the PM and sponsors like best goes against those standards, even though it's the best solution for this project and the company as a whole. Need to convince her that it is ok and that people still respect her work and position.

Lecture 10/11 (Project Scheduling):

- Understand the role of a project schedule
- Understand how to develop a project schedule
- Understand how to use a project schedule to monitor and track project progress
- Understand agile planning principles

- **What does the project schedule contain?**
  - Tasks that need to be completed
  - Duration and dependencies for each task
  - People and physical resources required by each task
  - Milestones and deliverable
  - Project Timeline

Project Planning and Scheduling Apply to Formal SDLC Processes – Plan Driven



Developing a Project Schedule:

1. Breakdown the task into small chunks you can deal with – Work Breakdown Structure (WBS)

2. Identify the interdependencies between the broken down tasks and develop a task network

3. Estimate the effort and the time allocation for each task

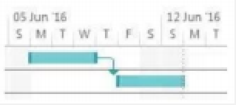
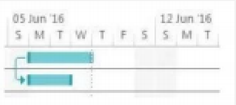
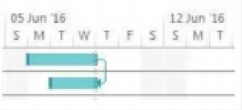
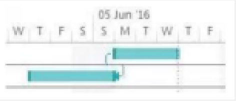
4. Allocate resources for tasks and validate effort

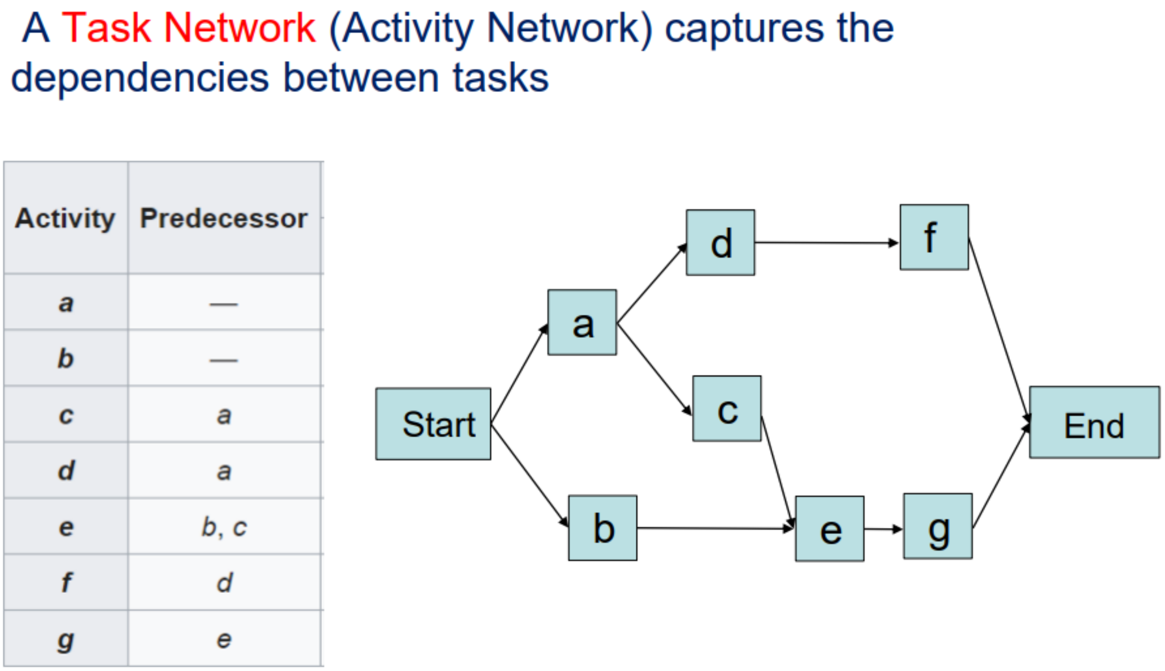
5. Develop the project schedule

Each Task should have a specific outcome ore deliverable.

1. **Breakdown the task into small chunks you can deal with. WBS tree.**

2. Identifying the Interdependencies between broken down tasks:

Dependency	Description	Representation
Finish-to-Start	Predecessor must finish before Successor can start	
Start-to-Start	Predecessor must start before Successor can start	
Finish-to-Finish	Predecessor must finish before the Successor can Finish	
Start-to-Finish	Predecessor must start before the Successor can finish	

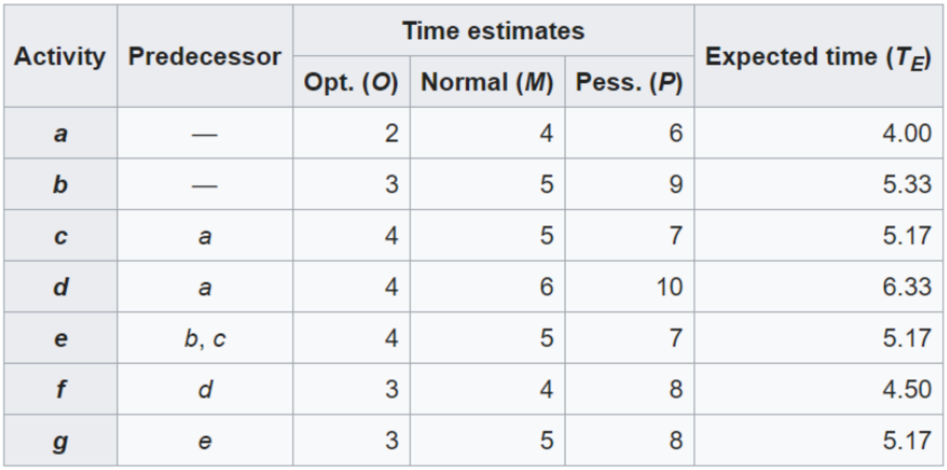


A Task Network (Activity Network) captures the dependencies between tasks

Activity	Predecessor
a	—
b	—
c	*a*
d	*a*
e	*b*, *c*
f	*d*
g	*e*

3. Estimate the effort and the time allocation for each task

Time estimation = (optimistic T + 4*Modest T + pessimistic T)/6



Activity	Predecessor	Time estimates			Expected time (T_E)
		Opt. (*O*)	Normal (*M*)	Pess. (*P*)	
a	—	2	4	6	4.00
b	—	3	5	9	5.33
c	*a*	4	5	7	5.17
d	*a*	4	6	10	6.33
e	*b*, *c*	4	5	7	5.17
f	*d*	3	4	8	4.50
g	*e*	3	5	8	5.17

4. Allocate Resources for tasks and validate effort

Care: Consider expertise of workers and availability when estimating resources.

5. Develop a Project Schedule

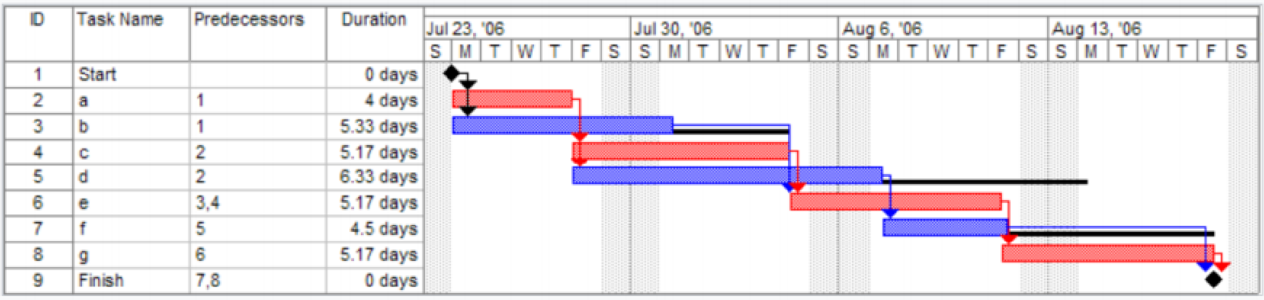
Project Schedule:

How long will it take to develop the system and how much will it cost?

Term	Description
Activity (Task)	Is part of a project that requires resources and time
Milestone	Is the completion of an activity that provides evidence of a deliverable completion or end of a phase – is an event that takes zero time
Free float (free slack)	Is the amount of time that a task can be delayed without causing a delay to subsequent tasks
Total float (total slack)	Is the amount of time that a task can be delayed without delaying project completion
Critical path	Is the longest possible continuous path taken from the initial event to the terminal event
Critical activity	Is an activity that has total float equal to zero

Charts for project schedule:

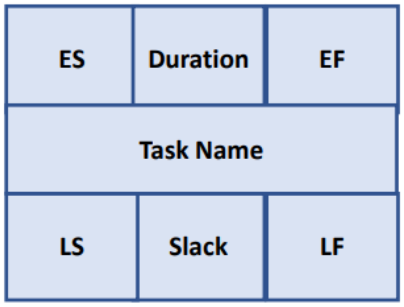
Gantt Chart: Bar chart showing schedule against calendar.



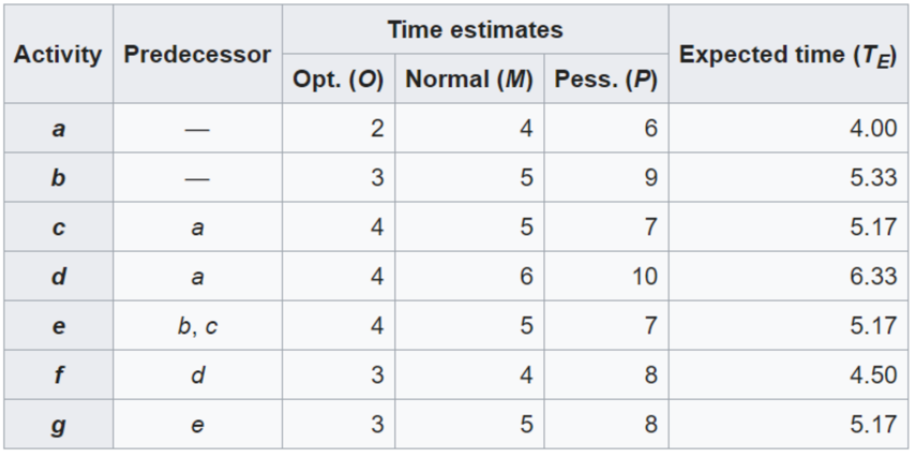
ID	Task Name	Predecessors	Duration
1	Start		0 days
2	a	1	4 days
3	b	1	5.33 days
4	c	2	5.17 days
5	d	2	6.33 days
6	e	3,4	5.17 days
7	f	5	4.5 days
8	g	6	5.17 days
9	Finish	7,8	0 days

A Gantt chart created using Microsoft Project (MSP). Note (1) the critical path is in red, (2) the slack is the black lines connected to non-critical activities, (3) since Saturday and Sunday are not work days and are thus excluded from the schedule, some bars on the Gantt chart are longer if they cut through a weekend.

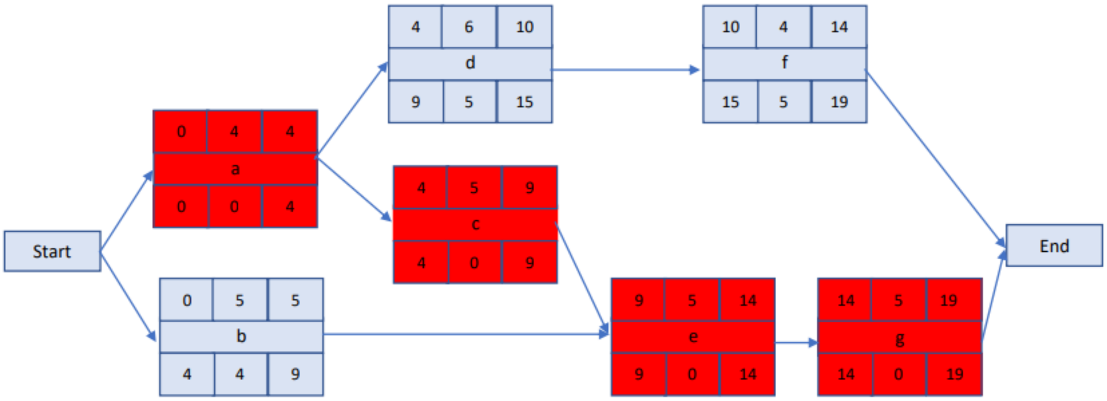
PERT Chart: Activity network showing dependencies among tasks and the critical path.



- Earliest start time (ES)
- Latest start time (LS)
- Earliest finish time (EF)
- Latest finish time (LF)
- Slack time

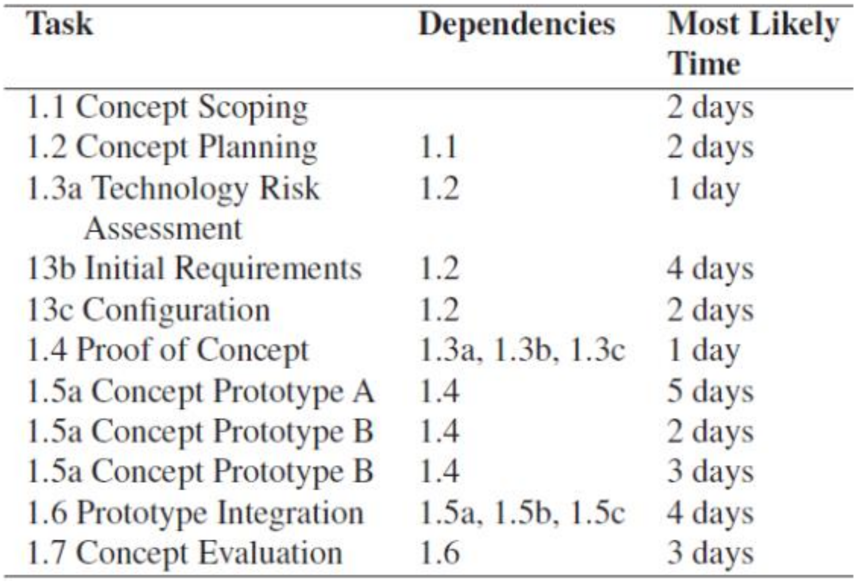


Activity	Predecessor	Time estimates			Expected time (T_E)
		Opt. (*O*)	Normal (*M*)	Pess. (*P*)	
a	—	2	4	6	4.00
b	—	3	5	9	5.33
c	*a*	4	5	7	5.17
d	*a*	4	6	10	6.33
e	*b, c*	4	5	7	5.17
f	*d*	3	4	8	4.50
g	*e*	3	5	8	5.17

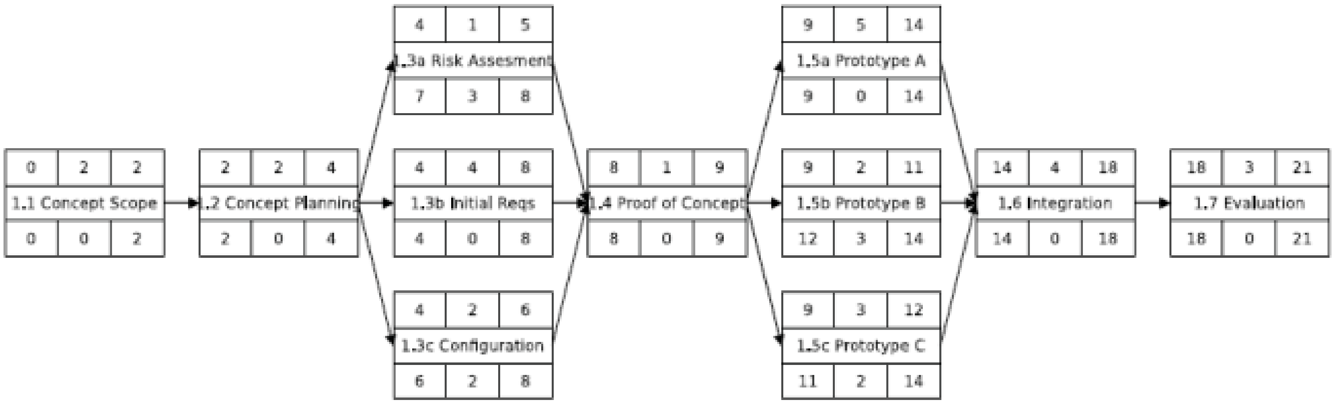


Critical Path: a, c, e, g
Duration: 19 days

Note: Critical path activities have a total free slack of 0



Task	Dependencies	Most Likely Time
1.1 Concept Scoping		2 days
1.2 Concept Planning	1.1	2 days
1.3a Technology Risk Assessment	1.2	1 day
13b Initial Requirements	1.2	4 days
13c Configuration	1.2	2 days
1.4 Proof of Concept	1.3a, 1.3b, 1.3c	1 day
1.5a Concept Prototype A	1.4	5 days
1.5a Concept Prototype B	1.4	2 days
1.5a Concept Prototype B	1.4	3 days
1.6 Prototype Integration	1.5a, 1.5b, 1.5c	4 days
1.7 Concept Evaluation	1.6	3 days



Critical Path: 1.1, 1.2, 1.3b, 1.4, 1.5a, 1.6, 1.7
Duration: 21 days

Note: Critical path activities have a total free slack of 0

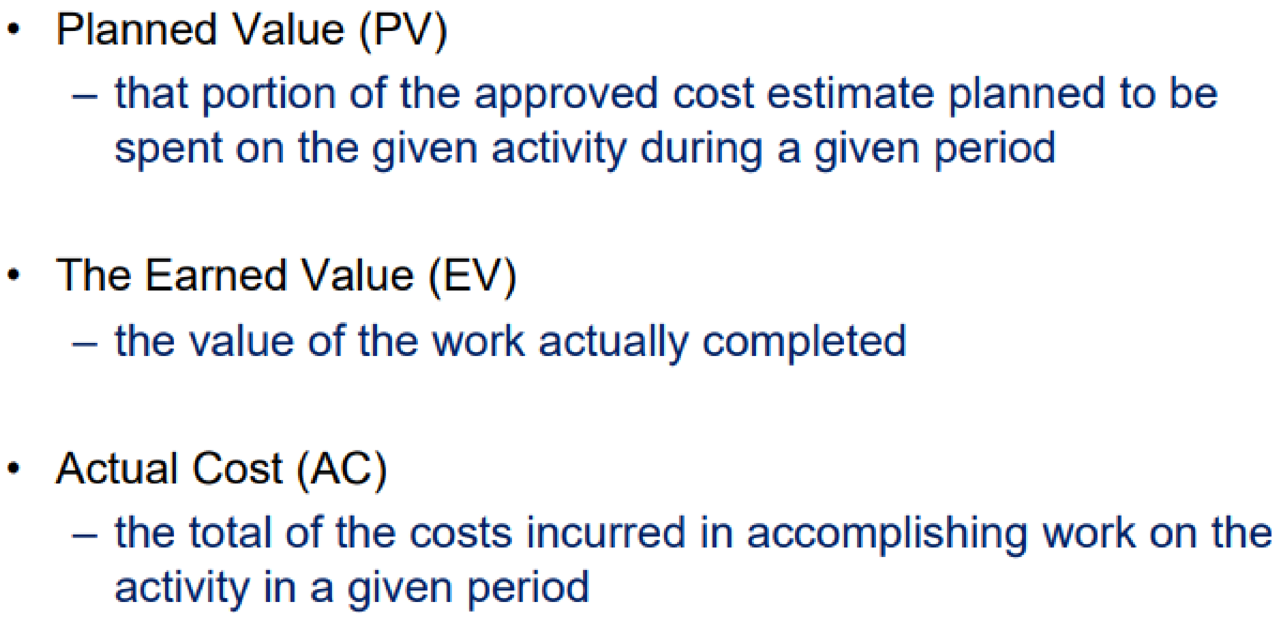
Critical Path: Path with longest duration. A delay in any of the activities in the critical path will cause the project to delay.

Crashing the Project Schedule: Shortening total duration by shortening critical path. (Removing dependencies between critical path activities, shortening duration of activities in critical path).

How to track and control project progress?

Earned Value Analysis (EVA)

- Report current/past project performance
- Predict future project performance based on current/past

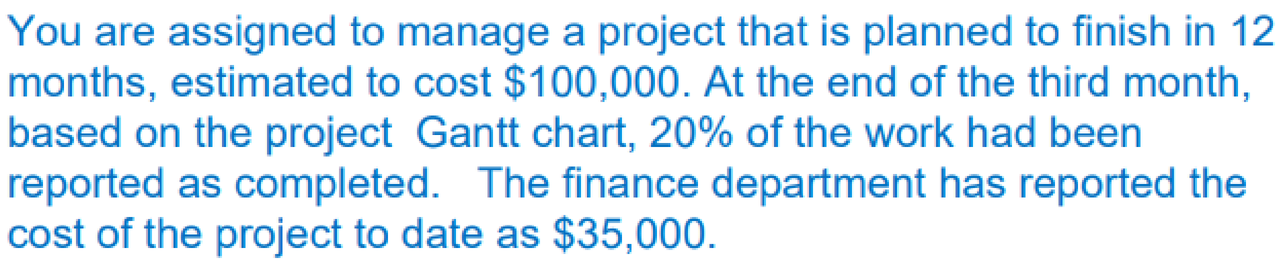


- Planned Value (PV)
  - that portion of the approved cost estimate planned to be spent on the given activity during a given period

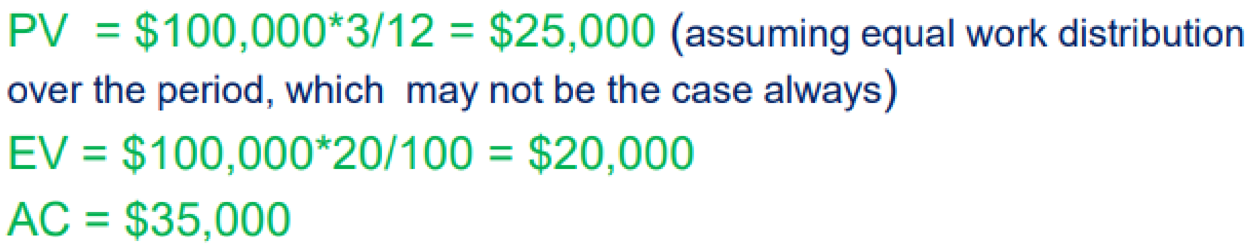
- The Earned Value (EV)
  - the value of the work actually completed

- Actual Cost (AC)
  - the total of the costs incurred in accomplishing work on the activity in a given period

Example:



You are assigned to manage a project that is planned to finish in 12 months, estimated to cost $100,000. At the end of the third month, based on the project Gantt chart, 20% of the work had been reported as completed. The finance department has reported the cost of the project to date as $35,000.



PV = $100,000*3/12 = $25,000 (assuming equal work distribution over the period, which may not be the case always)

EV = $100,000*20/100 = $20,000

AC = $35,000

- **Schedule Variance: expressed in dollars**

  $SV = EV - PV$

  $= 20{,}000 - 25{,}000$

  $= (5000)$

- **Schedule Performance Index: expressed as a fraction**

  $SPI = EV/PV$

  $= 20{,}000/25{,}000$

  $= 0.8$

Cost Variance: expressed in dollars

$CV = EV - AC$

$= 20{,}000 - 35{,}000$

$= (15{,}000)$

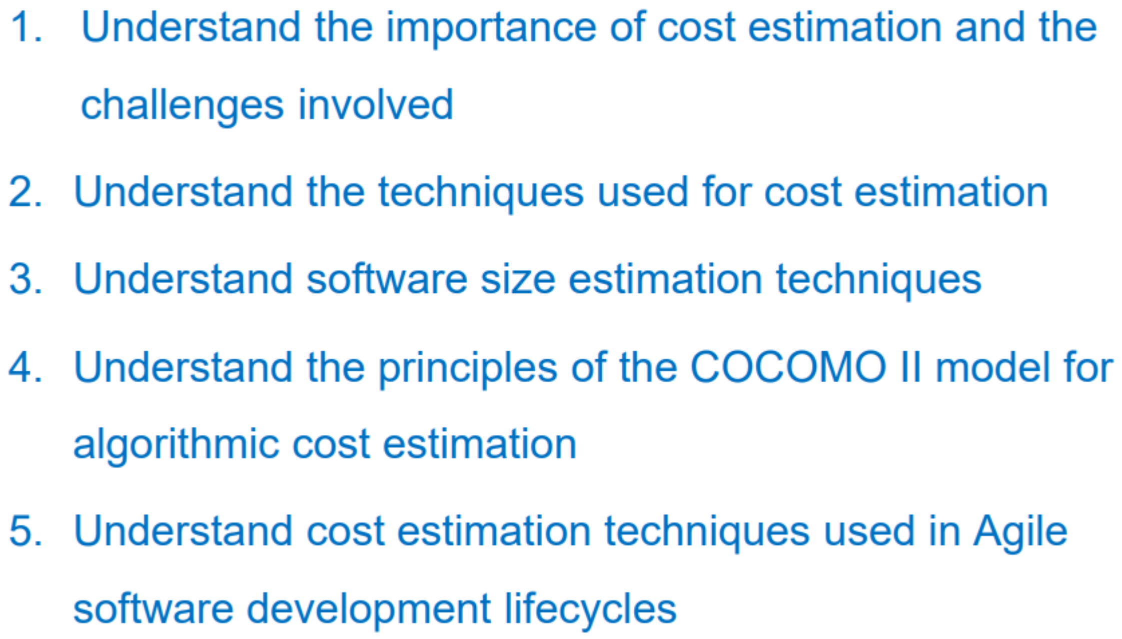
Cost Performance Index: expressed as a fraction

$CPI = EV/AC$

$= 20{,}000/35{,}000$

$= 0.57$

Lecture 12 & 13 (Cost Estimation):



1. Understand the importance of cost estimation and the challenges involved
2. Understand the techniques used for cost estimation
3. Understand software size estimation techniques
4. Understand the principles of the COCOMO II model for algorithmic cost estimation
5. Understand cost estimation techniques used in Agile software development lifecycles

What is Software Cost Estimation: Estimate how much money, effort, resources and time will take to build software system/product.

Why is it Important? Planning, is it feasible? How many workers do I need?

Techniques for Cost Estimation:

1. **Expert Judge:** Using experts to estimate the cost and reach a consensus.
2. **Estimation by Analogy:** Estimate based on similar project in same domain.
3. **Parkinson's Law:** Cost determined by available resources.
4. **Pricing to Win:** Cost depends on the budget not on software functionality.
5. **Algorithmic Cost Modelling:** Model created using historical information and some software metric. Size -> effort -> schedule -> cost

Software Size Estimation:

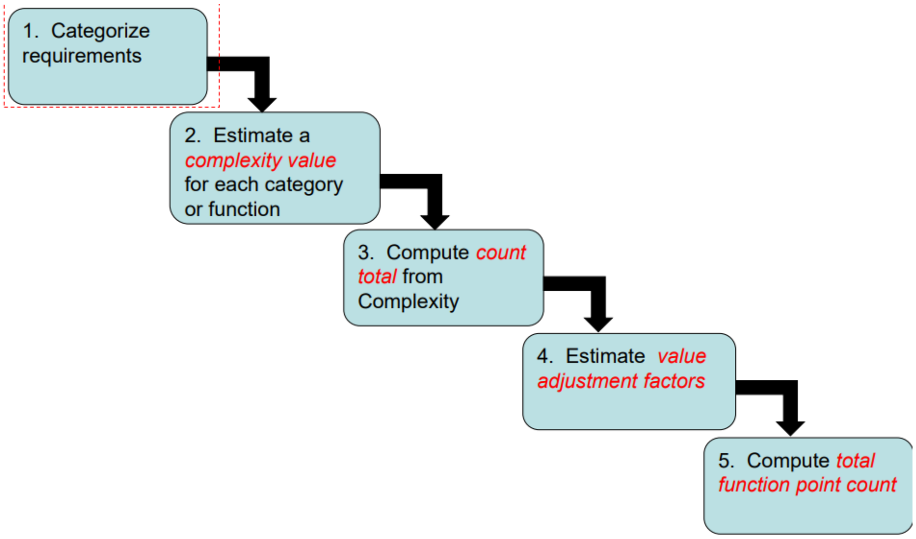
Source Lines of Code: Number of lines / Number of executable statements

Advantages: Can automate. Intuitive metric as it is visual.

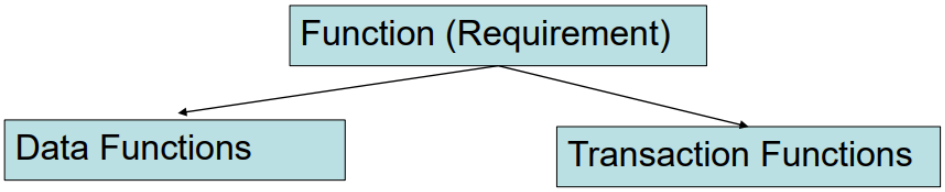
Disadvantages: Depends on programmer experience and programming language.

Function Points: Expresses functionality. Computed from (Software Requirements Specification (SRS)) which contains functional and non-functional requirements.

Function Point Computation Steps:


1. Categorize requirements
2. Estimate a *complexity value* for each category or function
3. Compute *count total* from Complexity
4. Estimate *value adjustment factors*
5. Compute *total function point count*

1. Categorize Requirements



ILF - tables in DB

EIF - External data

EI – Input from external

EO – output externally

EQ – input used to query

Data Functions:

- Concerned with maintenance of the data for the application – Internal Logical Files (ILF), External Interface Files (EIF)

Transaction Functions:

- Concerned with information being passes to and from the system - External Inputs (EI), External Output (EO), External Inquiries/Queries (EQ)

2. Estimate a complexity value for each category or function

Data, record (group of data) or file. Value of complexity: Simple/Average/Complex

3. Compute Count Total from Complexity

Using complexity and count estimate total count.

4. Estimate Value adjustment factors

Each characteristic is ranked on a scale of 0-5. 0 – Unimportant, 5 – Critical

5. Compute total function point count

Count total and value adjustment factors are then plugged into formula to estimate the total point count.

Advantages of Functional Pointers:

- Measure size of solution instead of size of problem
- Only need requirements, can be estimated early.
- Independent of technology and programming language.

Disadvantages of Functional Pointers:

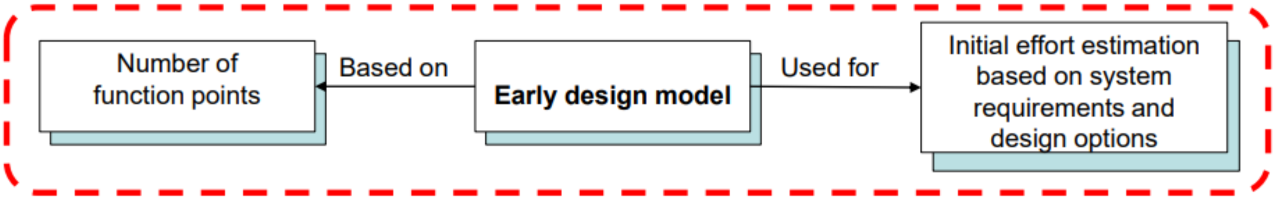
- Need well defined requirements
- Can be time consuming and costly

Use-case Points: measure software size using use-cases.

Advantages: Measured very early in project life cycle. Easy to use.

Disadvantages: Need requirements -> use cases. Need well-written use cases.

Understand the principles of the COCOMO 2 model for algorithmic estimation:


Number of function points
Based on
Early design model
Used for
Initial effort estimation based on system requirements and design options

- Derived from collection data from a large number of historical software projects and deriving formulae that best fit the observations. Effort based on size.

Understand the cost estimation techniques used in Agile SDLC:

Two key concepts used for effort estimation:

Story points: A relative measure of the size of a user story.

Velocity: Measure of the productivity of a team. Number of story points delivered in specified time period.

Agile Estimation Guidelines:

- Develop user stories for the system.
- Estimate number of story points for each story.
- Use team’s velocity from previous experience to estimate project delivery time.
- Re-estimate delivery time using velocity.

Agile Estimation Techniques:

Planning Poker:

- Customer reads story
- Team estimates
- Team discusses
- Team estimates again

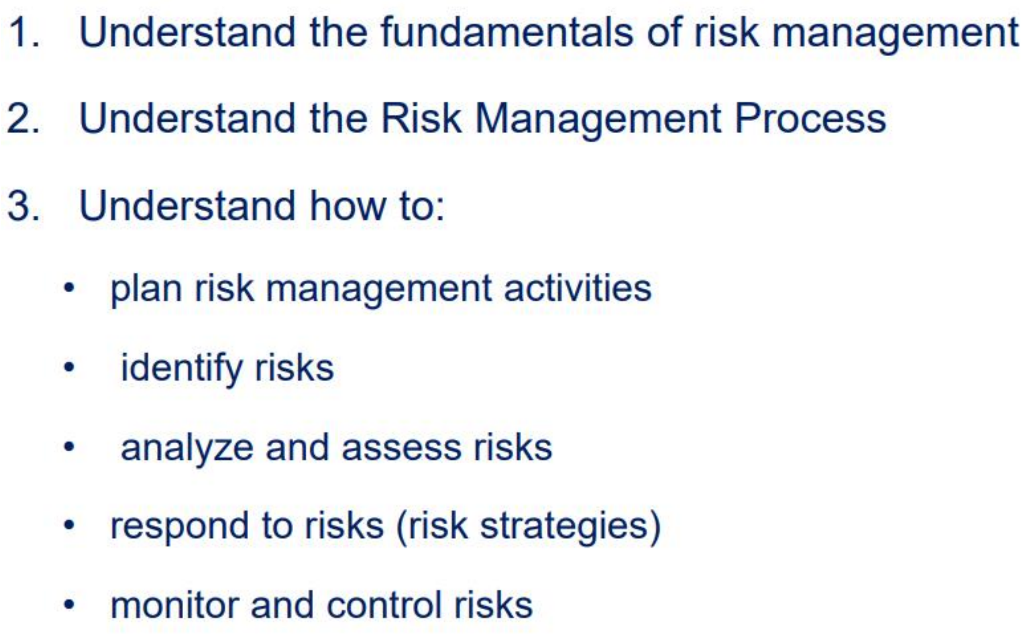
Bucket System:

- Team sitting at table picks a user story card randomly and puts in bucket 8.
- Next few cards randomly picked, discussed and agreed on based on previous ones and put in bucket.
- Then each person gets some cards and tries to balance them. (divide and conquer)
- Repeat until consensus

Relative Mass Valuation:

- Set up a large table so that stories can be moved around easily relative to each other.
- Pick if story is large, medium or small. Move to separate parts of table.
- Assign points values based on the position of the stories on the table.

Lecture 14 & 15 (Risk Management):



1. Understand the fundamentals of risk management
2. Understand the Risk Management Process
3. Understand how to:
    - plan risk management activities
    - identify risks
    - analyze and assess risks
    - respond to risks (risk strategies)
    - monitor and control risks

Risk: Uncertain event/condition with a positive or negative effect.

Uncertainty: Can cause a risk but not always.

Risk management important: Minimise impact of potential negative risks, maximize impact of potential positive risks. (**Probability**, **impact,** degree of **control, priority**)

Risk Management Process:

Plan -> Identify Risk -> Assess priority -> Decide action to reduce impact -> Monitor and control plan

Risk Management Plan: Documents the procedures for managing risks throughout a project.

Risk Identification Techniques:

Pondering: Sitting and thinking

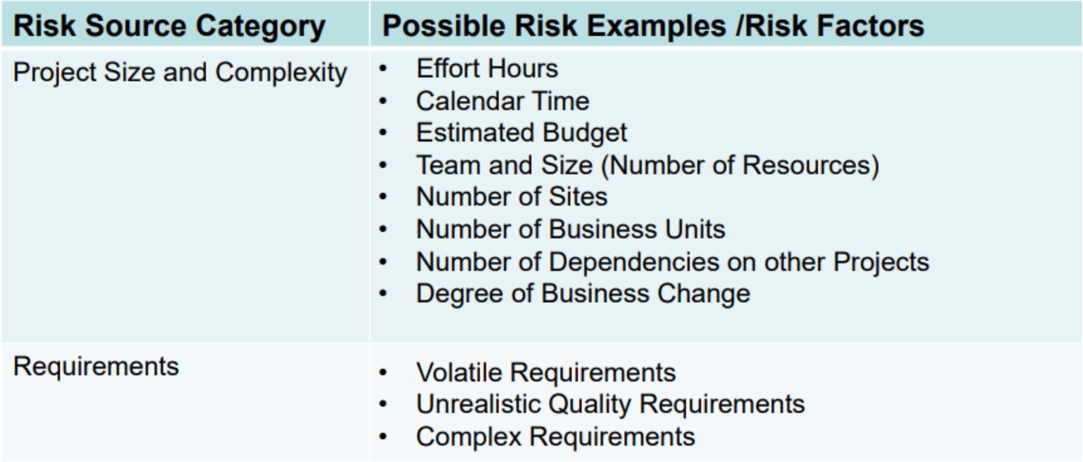
Interviews/Questionnaires: Interviewing stakeholders / domain experts

Brainstorming: Team contribution

Delphi Technique: Group of experts asked to identify risks and impact.

SWOT Analysis

Identified Risks Example:

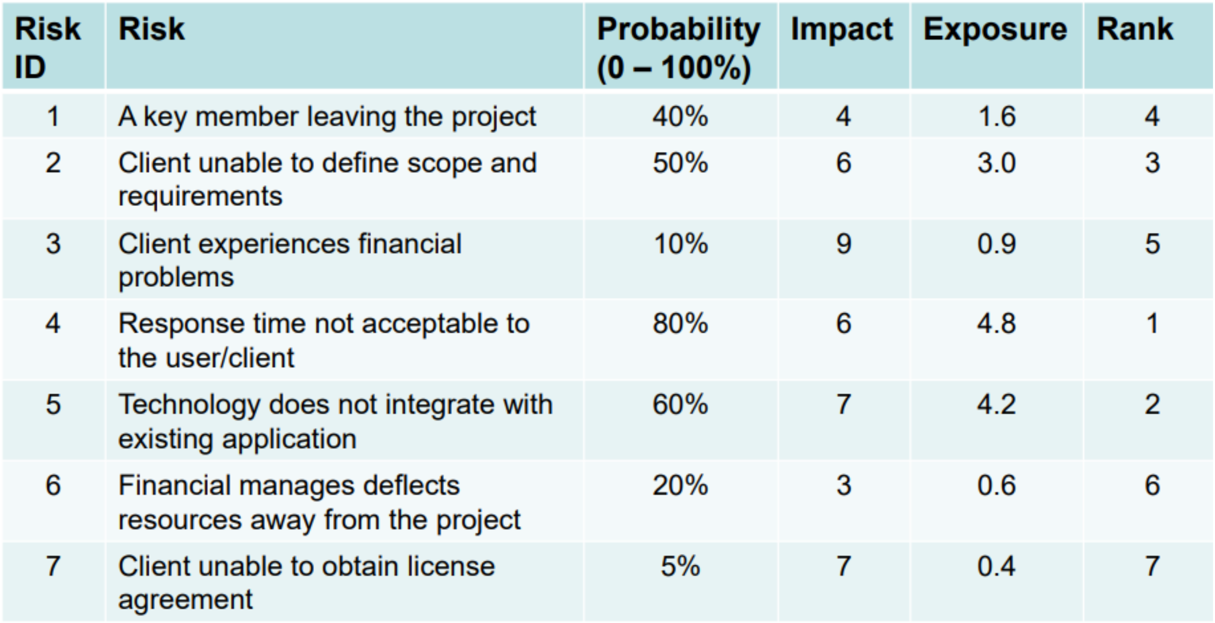


Risk Source Category	Possible Risk Examples /Risk Factors
Project Size and Complexity	• Effort Hours • Calendar Time • Estimated Budget • Team and Size (Number of Resources) • Number of Sites • Number of Business Units • Number of Dependencies on other Projects • Degree of Business Change
Requirements	• Volatile Requirements • Unrealistic Quality Requirements • Complex Requirements

Important Steps in Risk Analysis:

1. Estimate the Risk probability. (expert judgement)
2. Estimate the Risk impact. (scale of 1 – 5) 1 no impact, 5 catastrophic
3. Compute risk exposure = probability * impact
4. Identify the root cause -> can control

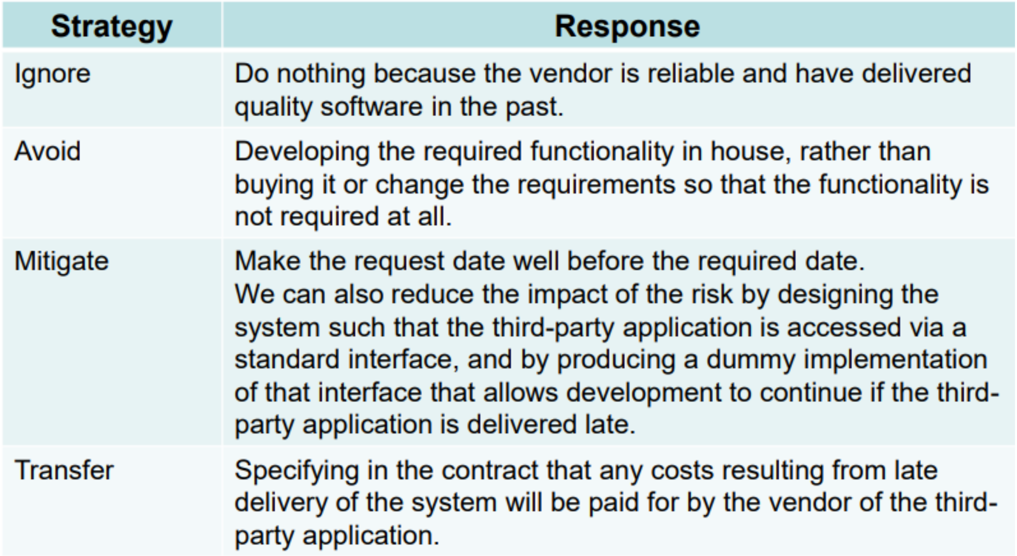
Risk Impact Analysis Table:



Risk ID	Risk	Probability (0 – 100%)	Impact	Exposure	Rank
1	A key member leaving the project	40%	4	1.6	4
2	Client unable to define scope and requirements	50%	6	3.0	3
3	Client experiences financial problems	10%	9	0.9	5
4	Response time not acceptable to the user/client	80%	6	4.8	1
5	Technology does not integrate with existing application	60%	7	4.2	2
6	Financial manages deflects resources away from the project	20%	3	0.6	6
7	Client unable to obtain license agreement	5%	7	0.4	7

Strategies to handle threats:

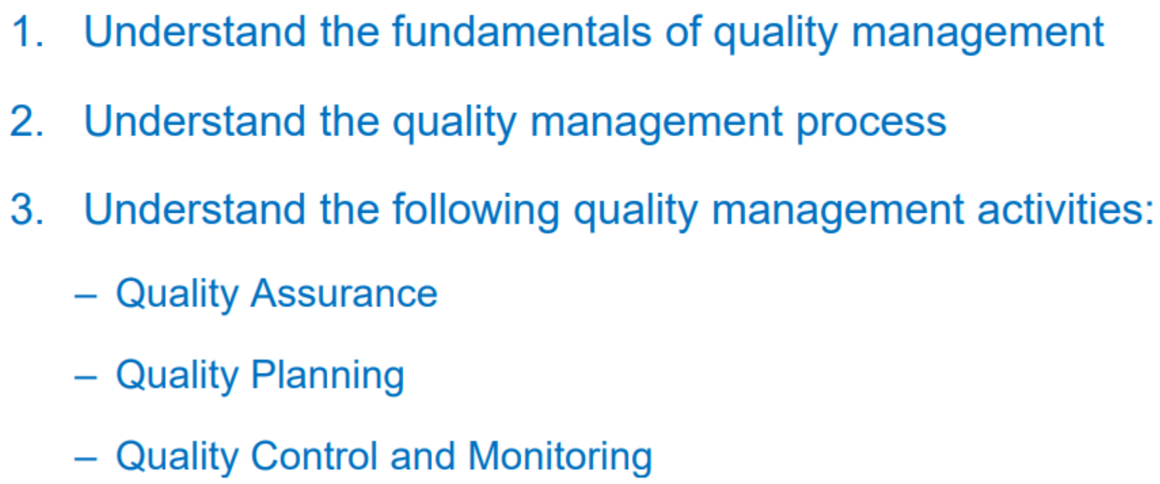
1. **Accept or ignore:** Not worth using resources
2. **Avoid:** Prevent risk such that Prob = 0 or Impact = 0
3. **Mitigate:** Employ technique to reduce probability or risk/reduce impact
4. **Transfer:** Transfer burden of risk to another party. Insurer, outsource



Strategy	Response
Ignore	Do nothing because the vendor is reliable and have delivered quality software in the past.
Avoid	Developing the required functionality in house, rather than buying it or change the requirements so that the functionality is not required at all.
Mitigate	Make the request date well before the required date. We can also reduce the impact of the risk by designing the system such that the third-party application is accessed via a standard interface, and by producing a dummy implementation of that interface that allows development to continue if the third-party application is delivered late.
Transfer	Specifying in the contract that any costs resulting from late delivery of the system will be paid for by the vendor of the third-party application.

Remember to monitor and control risk by having **risk reviews**, **risk audits** and **meetings**.

Lecture 16 & 17 (Quality Management):



1. Understand the fundamentals of quality management
2. Understand the quality management process
3. Understand the following quality management activities:
   - Quality Assurance
   - Quality Planning
   - Quality Control and Monitoring

Two broad perspectives of software Quality:

End-user’s Perspective: External quality characteristics. Reliability, performance, functionality, aesthetic.

Developer’s Perspective: Internal quality characteristics. Extendable, re-usable, low maintenance.

Quality important: Quality of the process impacts quality of the product. Impacts the satisfaction of the end user or client.

Cost of Quality: Quality assurances activities can be costly maybe have no assurance and then fix the faults (cheaper?). Weight up the opp and costs. E.g instead of doing formal review of requirements, build and then fix if client has problems.

Quality Management Process:

1. **Quality assurance:**

- Organizational procedures and standards for high quality software
- **Product standards:** Product being developed
- **Process standards:** Development process that should be followed

2. **Quality planning:** Select appropriate procedures and standards from the framework, adopted for the specific project.

Software Quality Assurance Plan: Product overview, Product plan (release dates), quality goals, process description, standards, risk management.

Verification: Ensures product meets requirements before final testing. Looking at processes or at code. Its static don't need to run anything, just read.

Validation: Unit testing, integration testing, acceptance testing. Dynamic.

Verification Processes:

Technical Reviews: Review of artefacts performed by peers. Author involved. Good for early detection of problems, may be time consuming. Informal: chat at the desk. Formal: multiple stakeholders developers/testers/client.

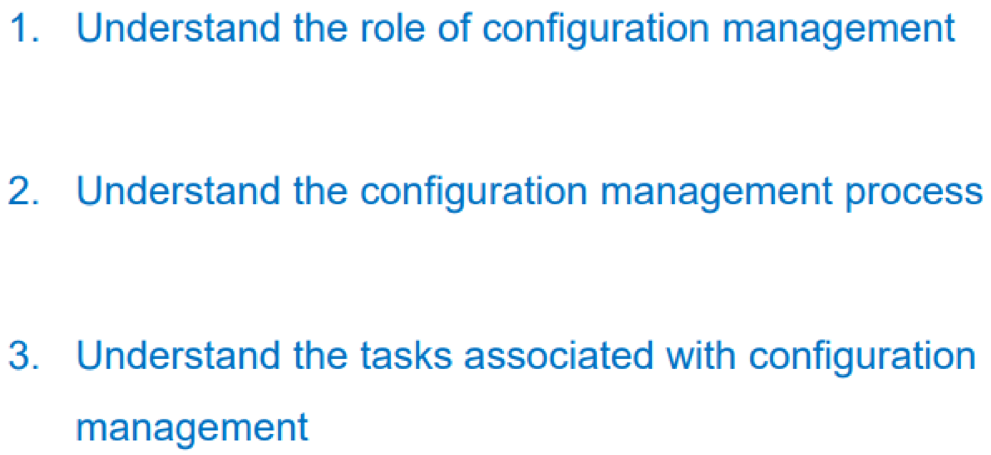
Audits: External party, author not involved.

Business Reviews: Functionality specified in scope is achieved.

Management Reviews: Compare progress against a baseline project plan. Resources need re-allocation? Project meets the scope?

3. **Quality control:** Ensure development team followed quality procedures and standards. As well as following the quality assurance plan.

Lecture 18 (Configuration Management):



1. Understand the role of configuration management

2. Understand the configuration management process

3. Understand the tasks associated with configuration management

Software configuration: Total of all artefacts, their state and dependencies between them.

Role of configuration management:

- Manage change without losing consistency
- Change can impact artefacts dependencies
- Establish processes
- Version control?

Configuration Management Process:

- Identify all the items that collective make up config
- Manage changes to items so that others remain consistent.
- Version control and quality control over time

Configuration Management Tasks:

Identification: Find artefacts: use-cases, user stories, requirements specification.

Version Control: Repository, version management function (git)

Change Control: Change management plan to control changes in configs. Handling. A **baseline** is an artefact that is stable.

Configuration auditing and reporting: Assuring repos are consistent. **Status reporting** is a way for large projects to keep track of change.

Lectures 20 & 21 - Tools

Confluence: Team collaboration software. Used for Documentation.

Jira: Sprint planning with help refocus execution, mitigate risks, guarantee higher quality code. Issue tracking project management functions.

Bitbucket: Version control, collaboration.

Bamboo: Bamboo is a continuous integration and continuous deployment server. Unit testing code commits, before publishing and performing live deployment.

HipChat: Internal private online chat and instant messaging.

Lectures 22 – Ethics and Outsourcing

L23.1 Understand how and why Ethics are important

L23.2 Australian Computer Society Code of Ethics

L23.3 Understand what Outsourcing is and why it is used

L23.4 Understand the types of contracts, when you would use them and key contractual clauses

L23.5 Understand the Procurement Management Process

Ethics:

- Sometimes doing what's ethical comes at a cost. Integrity. Image.
- Australian Computer Society Code of Ethics: Enhancement of Quality of Life. Honest. Professionalism.

Outsourcing: Engaging an external party to perform services or create goods that are traditionally performed in-house by the company's own employees.

Onshoring: Relocating activities inside national borders

Nearshoring: Near country e.g NZ

Offshoring: Another country or time-zone. E.g India

Advantages:

- Lower costs
- Access to more skills
- Save time

Disadvantages:

- Communications issues
- Loss of control
- Quality issues

Contracts:

One source of truth for all activities that are delivered by external parties. Mutually binding agreement. Clarifies responsibilities.

Fixed price contracts: Involve fixed to total price for well-defined product/service

Cost Reimbursable contracts: Payment to seller for direct and indirect costs. Usually an additional payment.

Time and Material contracts: Payment to the seller for actual time spent and any materials used in providing the service.

Fixed price has low buyer risk and his seller risk.

Contractual Clauses: Quality, Time, Location, Milestones, Fees, Intellectual property.

Procurement Management Process:

1. **Planning:** Consulting stakeholders to define the real need. Analyse how the supply market works. Assess risks.
2. **Sourcing:** Identify and engage with suppliers who will provide the best value for money outcome. Determine sourcing method like request for tender. We present a good statement of work (SOW) to give bidders understanding of what us as the buyers want.
3. **Managing:** Ensure rights and responsibilities are followed. Changes need to be approved. Determine if work is completed correctly.